REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
16-05-2012		

4. TITLE AND SUBTITLE

An Adaptive H∞ Control Algorithm for Jitter Control and Target Tracking in a Directed Energy Weapon

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Moran, Shane Christopher

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

Trident Scholar Report no. 407 (2012)

12. DISTRIBUTION/AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The project addresses two primary challenges of a directed energy weapon: platform jitter and target tracking. First to control platform jitter, an adaptive controller was created to attenuate tonal frequencies of the platform. With a stable beam, a second controller tracks and targets the laser onto a moving target at a distance of approximately five meters. The results show reduction in induced jitter by over 78%, showing a potential to significantly increase the effectiveness of directed energy weapons.

15. SUBJECT TERMS

Adaptive H-infinity, jitter control, optimal control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		121	19b. TELEPHONE NUMBER *(Include area code)*

A TRIDENT SCHOLAR PROJECT REPORT

NO. 407

An Adaptive H_∞ Control Algorithm for Jitter Control and Target Tracking in a Directed Energy Weapon

by
Midshipman 1/C Shane C. Moran, USN

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

USNA-1531-2

U.S.N.A. --- Trident Scholar project report; no. 407 (2012)

AN ADAPTIVE H_∞ CONTROL ALGORITHM FOR JITTER CONTROL AND TARGET TRACKING IN A DIRECTED ENERGY WEAPON

by
Midshipman 1/C Shane C. Moran
United States Naval Academy
Annapolis, Maryland

_____
(signature)
Certification of Adviser Approval

Professor R.T. O'Brien
Systems Engineering Department

_____
(signature)

_____
(date)
CAPT O.G. Thorp, USN
Systems Engineering Department

_____
(signature)

_____
(date)
CDR R.J. Watkins, USN
Mechanical Engineering Department

_____
(signature)

_____
(date)
Acceptance for the Trident Scholar Committee

Professor Carl E. Wick
Associate Director of Midshipman Research

_____
(signature)

_____
(date)

USNA-1531-2

ABSTRACT

In recent years the Office of Naval Research has undertaken the challenges posed by directed energy weapons with the creation of the Directed Energy Weapons Program. This program is aimed at developing more accurate and efficient directed energy. The program identified five main fields of focus necessary for creating an effective directed energy weapons system. A laser travels at the speed of light, redefining the type of targeting and tracking method used. It must remain on target, at a precise location, for the entire duration of fire to achieve maximum effectiveness. Directed energy weapons, like all mechanical systems, are subject to vibrations which cause the beam to deviate. With on-target precision being such an important aspect, the slightest vibrations in the directed energy system can cause tremendous problems. Accuracy is needed at the microradian level. This project will address two primary challenges of a directed energy weapon: platform jitter and target tracking. First, to control platform jitter, an adaptive controller was created to actively identify and attenuate tonal frequencies of the platform. With a stable beam, a second controller tracks and targets the laser onto a moving target at distance of approximately 5 m. The results show reduction in induced jitter by over ***78%,*** showing a potential to significantly increase the effectiveness of directed energy weapons.

Keywords: Adaptive H-infinity, jitter control, optimal control

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Motivation

In recent years the Office of Naval Research (ONR) has undertaken the challenges posed by directed energy weapons with the creation of the Directed Energy Weapon Program. This program is developing a directed energy laser weapon that will increase the Navy's effectiveness at shooting down enemy weapons and/or hostile craft while minimizing collateral damage. The program also designated five fields of focus: (1) free electron laser weapon system, (2) free electron laser for weapons of mass destruction detection, (3) high power microwave weapon, (4) electric fiber weapon system, and (5) beam control. Beam control for directed energy weapons, like any weapon, is necessary to correctly aim and fire the weapon in a combat environment. The primary difference from conventional weapons is that directed energy weapons travel at the speed of light and requires a new approach to aiming and tracking. To be effective, the directed energy laser has to be pointed on the target at all times and remain at the same location on the target to maximize the energy density delivered. Since the weapon remains connected to the launching platform, any disturbances acting on to the platform are reflected in the beam's motion on the target. To maintain a high energy density, the beam's path will be adjusted before it leaves the platform, counteracting the induced jitter. Another factor affecting the precision of this weapon is the propagation through the atmosphere; however, this issue is not addressed in this project. Researchers at both the Naval Post Graduate School (NPS) and the United States Naval Academy (USNA), through support and tasking of ONR, are searching for possible solutions to the beam control issues.

With on-target precision presenting such an important aspect, the slightest vibrations in the directed energy system can cause tremendous problems. Directed energy weapons, like all mechanical systems, are subject to vibrations that cause the beam to deviate. To destroy a target, approximately 100 kW or more is needed, which is already a large amount without accounting for any loss in intensity. Even a deviation of 1 microradian, for a beam of 1cm diameter at 10 km, can cause a decrease in intensity of 9 times. Increasing the laser output, or time-on-target, by 900% is not a logical or even a feasible scenario with current technology. Research of optical beam control started with the advent of satellites, and with them, the idea of directed energy

weapons. But only recently have we developed the technology to produce lasers that are small, portable and powerful enough for real world application. USNA, under the supervision of CDR Watkins, has developed a Directed Energy Beam Control Laboratory. This research project will develop and implement an adaptive control algorithm for jitter control and target tracking on a directed energy beam.

1.2 Background

Vibration and disturbance rejection has been studied since the early 1900's. In depth analysis of jitter reduction in relation to optical beams started with the advent of laser satellite communications in the 1980's and 1990's[1]. Free-space laser communication provides many advantages to traditional microwave communication: broader bandwidth, lower power consumption, and higher security[2]. As a natural progression of theory, adaptive control techniques have been suggested for more accurate beam control[3]. At the time optical beam control was not deeply associated with a weaponized variant of directed energy, such as Regan's "Star Wars" program, due to large size and lack of efficiency of current technology after initial review.

In the 1990's the Air Force designed and produced a high-altitude directed energy weapon, carried in a Boeing 747, which had the possibility of shooting down ballistic missiles, known as the Airborne Laser (ABL)[4]. While the ABL was a great step towards directed energy weapons, it avoided many of the dominating effects of a maritime environment. Flying at an

---

[1] Skormin, V.A.; Tascillo, M.A.; Nicholson, D.J., "A jitter rejection technique in a satellite-based laser communication system," Aerospace and Electronics Conference, 1993. NAECON 1993., Proceedings of the IEEE 1993 National , vol., no., pp.1107-1115 vol.2, 24-28 May 1993.

[2]Zhaowei Sun; Xiangzhi Li; , "Research on adaptive control algorithm of jitter in laser beam pointing and tracking system," Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on , vol., no., pp.893-897, 17-19 Dec. 2010

[3] Skormin, V.A.; Busch, T.E.; Givens, M.A., "Model reference control of a fast steering mirror of a pointing, acquisition and tracking system for laser communications," Aerospace and Electronics Conference, 1995. NAECON 1995., Proceedings of the IEEE 1995 National , vol.2, no., pp.907-913 vol.2, 22-26 May 1995

[4] Forden, G.E., "The airborne laser," Spectrum, IEEE , vol.34, no.9, pp.40-49, Sep 1997

altitude of 40,000 ft. the atmosphere was much clearer and the plane experienced a much less turbulent setting. For a maritime application there are more interfering particles in the atmosphere and the combination of sea states and ship structures create more turbulence and jitter. With recent advances in laser technology allowing for smaller and more powerful systems, the Navy has developed an interest in designing a maritime capable, directed energy weapons system. Though power output has increased, it is still at a point where accurate beam control is needed to achieve effective damage on a target. It is necessary for the Navy to develop an optical beam controller capable of reducing the effects of jitter substantially, in order to advance to the next step.

After tasking from ONR, previous work at USNA in the Directed Energy Beam Control Laboratory has looked at initial control design concepts for jitter control. In 2010 Ensign Shreffler (USNA Class of 2010) implemented a linear quadratic regulator and achieved approximately 65% attenuation. In 2011 Ensign Malinoski (USNA Class of 2011) implemented a pre-calculated H_∞ controller and achieved an attenuation of approximately 80%. The previous methods both implemented a controller that was pre-calculated using known frequencies of disturbance and a stationary target. The next step is to implement the successful H_∞ controller concept of Malinoski, but with the ability to identify and retune for different frequencies in real-time, while also tracking a moving target.

2. EXPERIMENTAL SETUP AND PROCEDURE

2.1 Description of Major Components

Research for this project was conducted in the USNA Directed Energy Control Laboratory as seen in Figure 1. The laboratory is located in Rickover Hall and is configured for safe use of Class IV lasers.

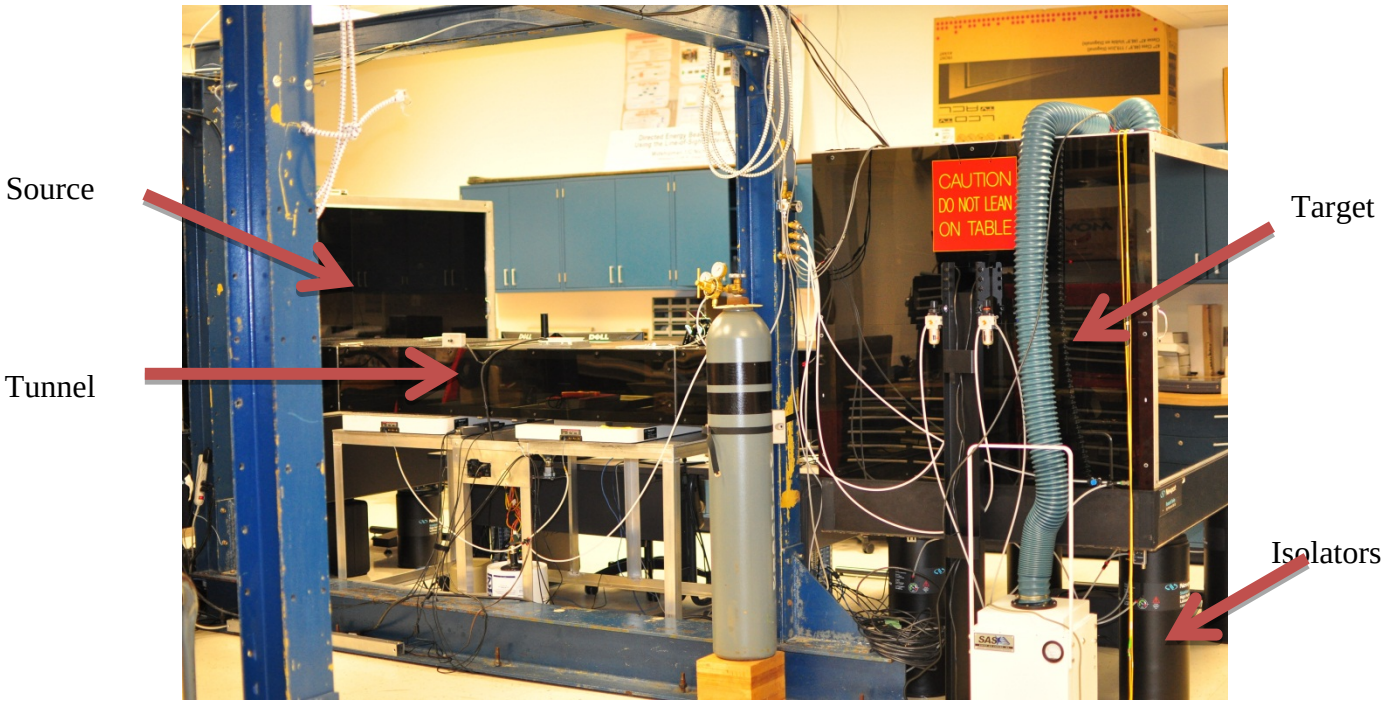


Figure 1 USNA Directed Energy Control Laboratory

This research uses the lab configuration diagramed in Figure 2 and the laboratory pictures in Figure 3 and Figure 4. Additional technical information for the hardware used while conducting this research in the Directed Energy Research Center can be found in the appendices.

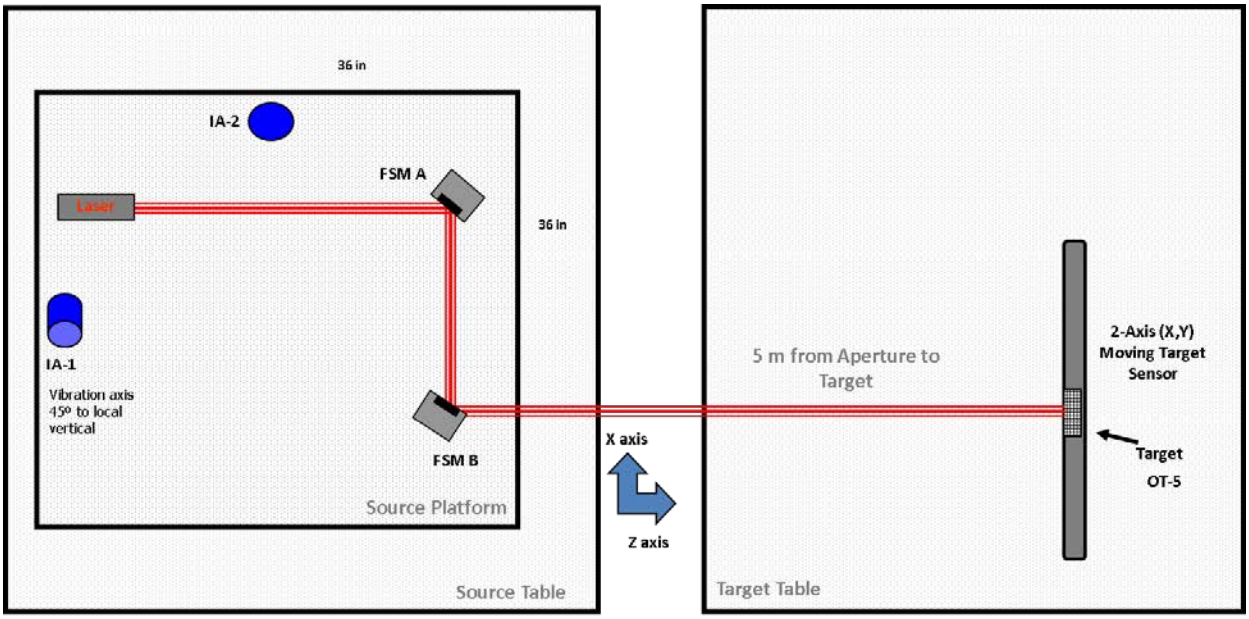


Figure 2 Research Configuration Schematic

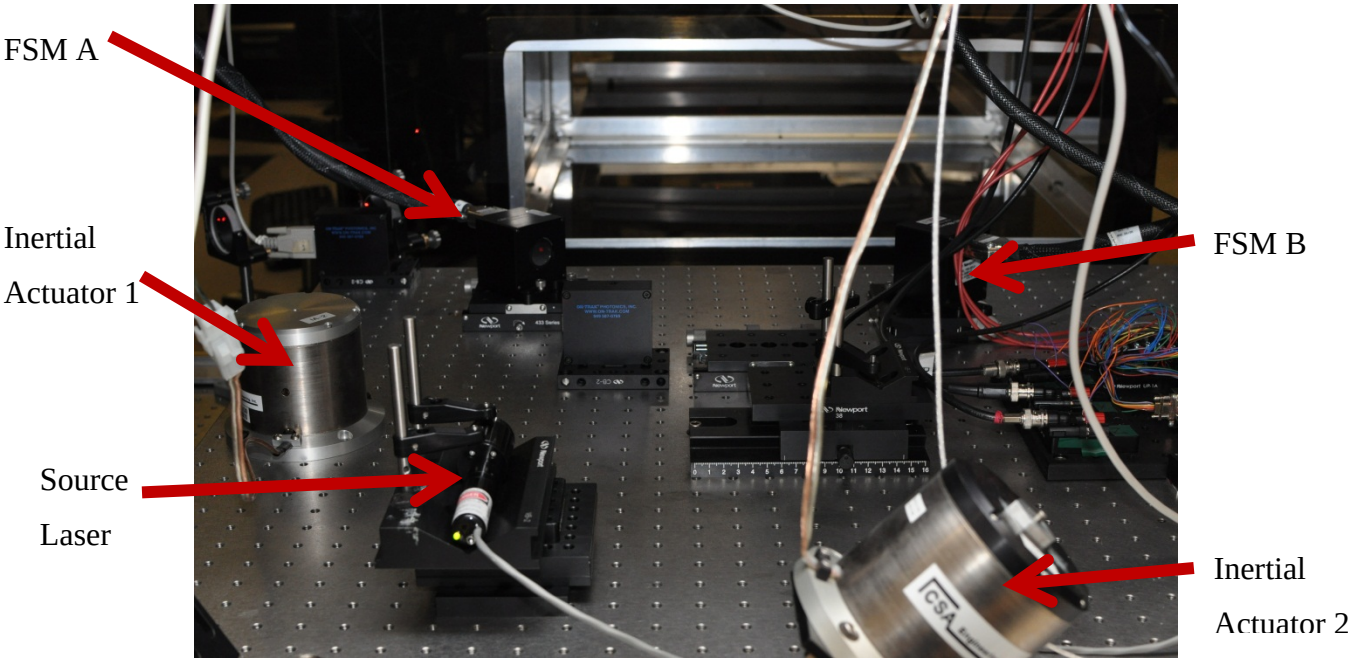


Figure 3 Source Platform Configuration

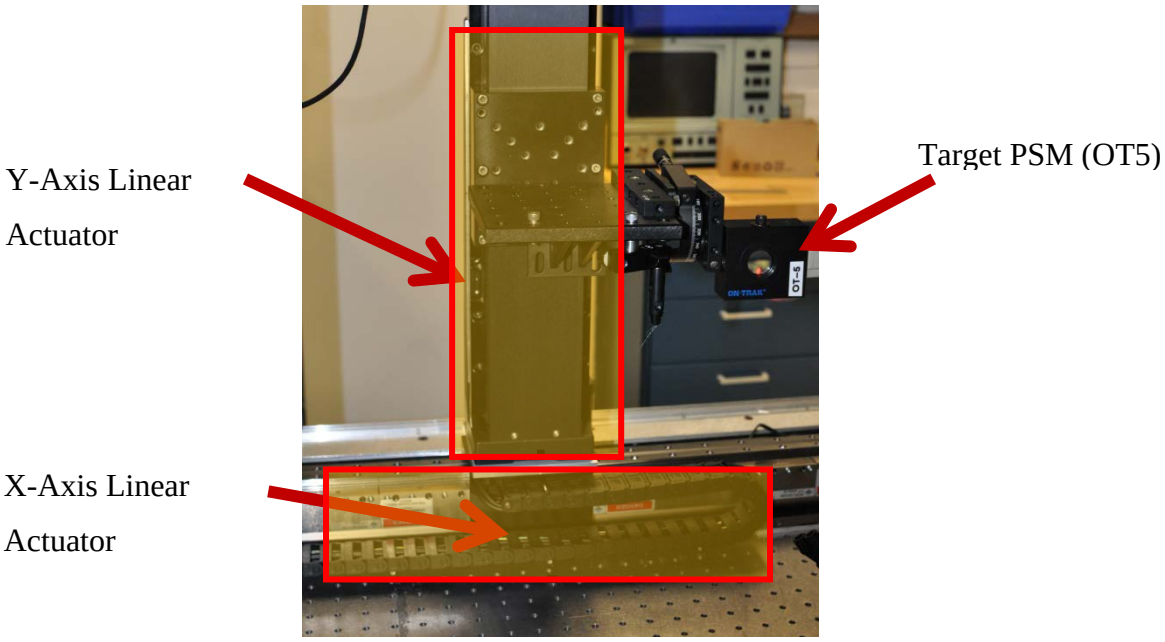


Figure 4 Target Platform and Sensor Configuration

The platform reference frame used in this research is defined with the positive z-axis being downrange from source to target, the positive y-axis up at vertical from the platform, and positive x-axis to the left of the platform at a horizontal to the surface. Roll, pitch, and yaw are all defined using the right-hand method; positive yaw in the positive y-direction, positive roll in the positive z-direction, and positive pitch in the positive x-direction.

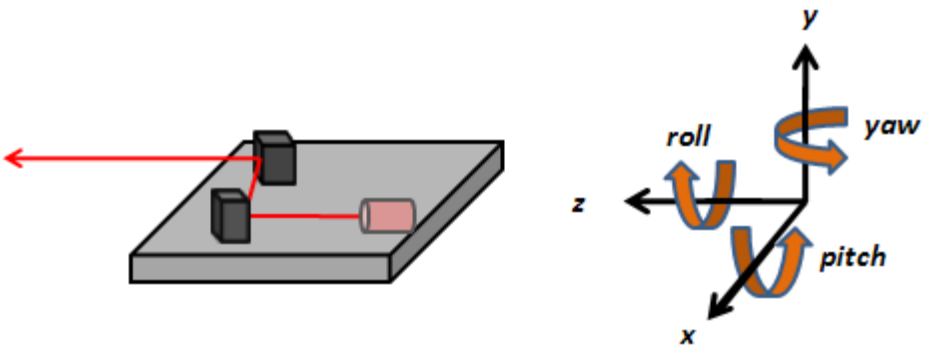


Figure 5 Platform Axis System

2.1.1 Position Sensing Module

On-Trak PSM 2-10 Position Sensing Modules (PSMs) shown below are each composed of a quadrilateral Position Sensing Detector (PSD) semiconductor chip connected to an On-Trak OT301 position sensing amplifier. PSMs detect the geometric centroid of the irradiance incident on the semiconductor face. The PSMs have a detection area of 10 mm x 10 mm and provide the position of the center of the laser beam in two dimensions. The minimum resolution of the PSM is approximately 0.5 micrometers when combined with the OT301 amplifier[5]. PSMs are used to determine the position and orientation of the platform in the off-platform motion sensing configuration and to determine the beam's position on the target.

[5]On-Trak Photonics Inc., "Position Sensing Modules-Position Sensing Instruments." 23 April 2011. <http://www.on-trak.com/psm/html>.

Figure 6 Position Sensing Module

2.1.2 Fast Steering Mirror (FSM)

The Fast Steering Mirrors (FSMs), produced by Newport Corporation, are the foundation of this control system. The FSMs allow the operator to both precisely and accurately steer the beam downrange. A one inch diameter mirror FSM-300 is used for jitter mitigation and named FSMA. A two inch diameter mirror FSM-320 is used for target tracking and named FSMB. FSMs offer high bandwidth, sub-microradian resolution, and two-axis "tip-tilt" rotation control using four voice coil actuators acting in push pull pairs. FSMA has a control bandwidth of 800 Hz making it suitable for jitter correction and FSMB has a bandwidth of 350 Hz, making it suitable for target tracking.[6] A FSM-300 (left) and a FSM-320 (right) are shown in Figure 7.

Figure 7 Newport FSM

2.1.3 Laser

---

[6] Newport Corporation, "Fast Steering Mirrors." 19 December 2009.

<http://www.newport.com/Fast-Steering-Mirrors/847119/1033/catalog.aspx>.

The Stocker Yale Canada Inc. Lasiris™ laser used on the platform for the source Lasiris a 5mW, 635 nm diode lasers, with an elliptical beam measuring 3.8 mm x 0.9 mm. The source laser is circularized by an Edmund Optics NT47-274 anamorphic prism pair beam expander, with a resulting beam diameter of 3.8 mm. The source laser is shown in Figure 8.

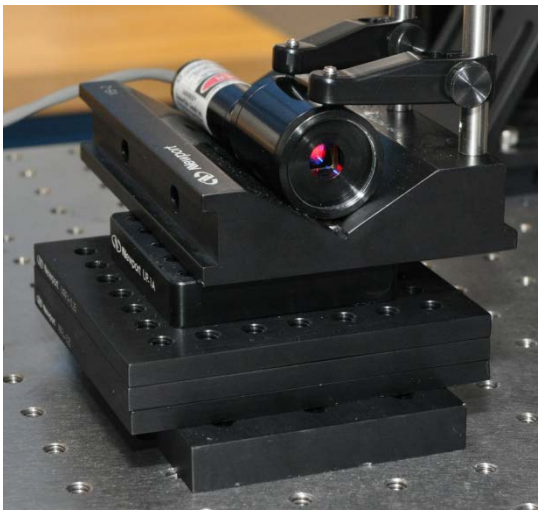


Figure 8 Source Laser with Beam Expander

2.1.4 Inertial Actuators

Disturbances are created by two CSA Engineering SA-10 Inertial Actuators mounted on the source platform with rated force outputs of up to 10 lb_f for frequencies up to 1,000 Hz. Inertial actuator 1 (IA1) is mounted at a 45 degree angle to the local vertical at the aft portion of the platform so as to impart both a pitch and a yaw motion. Inertial actuator 2 (IA2) is mounted vertically as shown in Figure 2 so as to impart a rolling motion to the platform. Input current for each individual frequency is limited to 3A, and a natural frequency of 30Hz is avoided as well. IA1 (left) and IA2 (right) are shown in Figure 9.

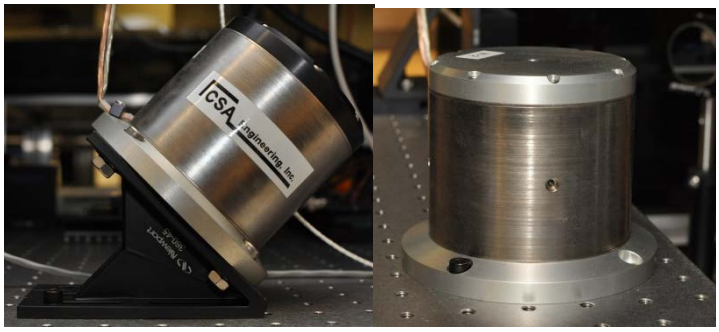


Figure 9 CSA Inertial Actuators

2.1.5 Breadboard

A Newport RG-33-2-ML research grade breadboard (Figure 10) acts as the directed energy beam's source platform. The breadboard measures 91.44 by 91.44 by 5.8 cm (36 by 36 by 2 in) and is constructed in a honeycomb pattern to eliminate torsional and bending modes below approximately 200 Hz. The mass of the bread board is 71.3 kg.

Figure 10 Newport Breadboard[7]

2.1.6 Isolation System

The source table (Newport RS2000-48-18) and target table (Newport RS4000-48-8) optical tables are isolated from the ground by Newport I-2000 Pneumatic Isolators with automatic leveling. The source table is 4 ft. x 8 ft. x 18 in. and the target table is 4 ft. x 8 ft. x 8 in. The source laser

[7] Newport Corporation, "Optical Breadboard."

platform breadboard is isolated from the optical table by four springs and four pneumatic isolators (Figure 11). The stainless steel springs are approximately 3.8 cm long with an outer diameter of 2.8 cm and a stiffness of 20 kN/m. The pneumatic isolators are Newport SLM-3A air mounts and are pressurized to 275 kPa resulting in a natural frequency of 3.5 Hz.[8]

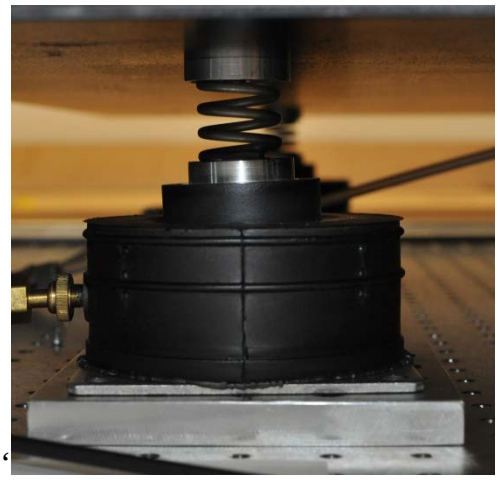


Figure 11 Spring-Pneumatic Isolators

2.1.7 Linear Motor Actuators

Aerotech model LMAC-095R-635 Linear Motor Actuators are controlled independently on each axis by an Aerotech Soloist™ dual-axis motion controller capable of moving the target PSM 635 mm with an accuracy of ±1 μm and a repeatability of ±0.5 μm at speeds up to 5 m/s.[9]

---

[8] Newport Corporation, "Compact Air Mount." 23 April 2011.

<http://search.newport.com/?x2=sku&q2=SLM-1A>.

[9]Aerotech Inc., "LMA/LMAC Series Linear Motor Actuators." 23 April 2011.

<http://www.aerotech.com/products/pdf/lms.pdf>.

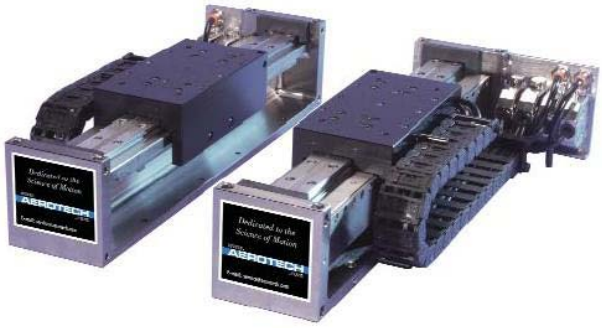


Figure 12 Aerotech Single-Axis Linear Motion Actuators

2.1.8 Computer System and Software

The control system was developed using Mathworks MATLAB R2011a with SIMULINK version 7.5 (R2011a) and experiments were run using the SpeedGoat xPC Targetbox. The main computers for control implementation and experiment supervision are two Dell T7400 work stations running Microsoft 7 on Intel Xeon CPUs with clock speeds of 3.40 GHz and 3.00 GB RAM each. The xPC Targetbox is an Intel Core 2 Duo running at 2.13 GHz.

Figure 13 Control System Workstation

2.2 **Experimental Method**

2.2.1 Experimental Assumptions

In this research, the platform is assumed to act as a rigid body. The source platform is designed to remain rigid below 200 Hz and the inertial actuators are not operated above 60 Hz. Jitter due to source platform translational movement along an axis is assumed to be minimal, and if significant, it will be detected by the sensor as target movement.

The angular velocity of the target relative to the source laser platform is assumed to be small. The angular motion of the target is described by equation 1 where *r* is the range to the target, *v* is the target velocity, and α is the angular rate of motion of the target relative to the platform.

$$v = r * sec^2(\alpha) \tag{1}$$

At tactical ranges of 10 km, a target moving rapidly at 400 kph represents only a small 0.64°/s rate of relative angular motion. Therefore, a relatively slow moving target at the 5 m range of the lab is sufficient to scale to a real target application.

2.2.2 Beam Control System

The laboratory components are arranged as shown below in the schematic in Figure 14.

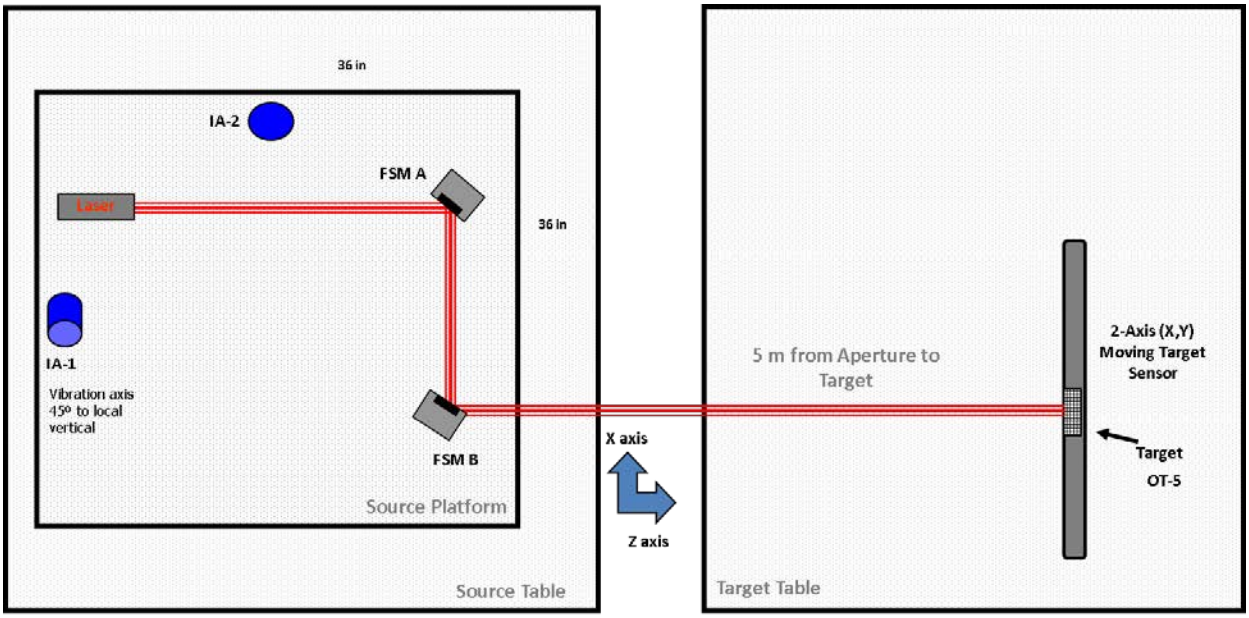


Figure 14 Experiment Configuration Schematic

The Newport breadboard is mounted on the spring-pneumatic isolator system which gives the board six limited degrees of freedom. The source laser is mounted on the source platform pointing down the positive z-axis. Along the beam path are two FSM's each mounted at a 45 degree angle to steer the beam. FSM A is mounted at a 45 degree angle to the source laser and is used to provide jitter control, thus providing a stable beam. FSM B is mounted at 45 degrees to the beam coming off FSM A and is used to provide target tracking motion.

On the target platform are two linear motion actuators, one mounted to the platform and one mounted on the other, allowing for independent axis control in both x-direction and y-direction. A PSM (OT-5) is mounted 4.81m in the z-direction from the face of FSM B. The PSM is mounted on two linear actuators that allow for independent control of both the x-axis and y-axis, while the z-axis distance remains the same. OT-5 provides a relative position of beam center on the detector. Two voltage outputs are given corresponding to x-axis location and y-axis location.

With the PSM at the target (OT-5) feeding back the relative beam position, data can be extracted in two independent loops for target tracking and jitter control. The goal is for the first loop to remove the higher frequency jitter, providing a tight beam to the second control loop, which will use the steady state error (lower frequencies) to track the target. The first control loop is for jitter mitigation and is used to control FSM A. Data from OT-5 is fed through a high-pass filter, allowing frequencies from 6Hz to 500Hz pass through, and is then used to identify the maximum frequencies of disturbance. After the tonal frequencies are identified they are fed into the H_∞ controller which commands FSM A and feeds a steady beam to FSM B. The second control loop takes the data from OT-5 and sends it through a low-pass filter, allowing frequencies from 2Hz to 8Hz to pass through. The target tracking controller uses two proportional-integral (PI) controllers in series to handle the ramp-up when the velocity profile changes. In both control loops there is a desired beam position defined as (0,0) on OT-5, and both control loops seek to minimize that error in two separate frequency spectrums. The system requires a calibration sequence so that the desired center position on the target is defined for the controllers and error feedback . All platform vibration and target movement are defined relative to this zeroed point on the target, OT-5. Once the platform is floated and beam is steadied, the location of the beam on OT-5 is measured, then electronically subtracted from both x and y-axes in the control algorithm.

2.2.3 Experimental Procedure

Experiments were run in three separate groups, the first two sets testing the individual subsystems, then the third testing the entire beam control system. The first two sets tested the target tracking and jitter control subsystems independently to ensure their stability. All experiments operated at sample rate of 1 kHz (.001 second time step). For cases when jitter control was tested, IA1 and IA2 produced disturbances of 12Hz and 38Hz beginning at 1 second after run start, and then shifted to 9Hz and 46Hz on both actuators at 7 seconds after run start. The jitter controller system starts at 1 second after run start and initiates the first controller at 2.024 seconds after run start, once it has identified initial frequencies. For cases when the target tracking system is tested, the target tracking controller turns on at 2.5 seconds after the run start.

Once the target tracking controller has locked onto the desired position the target motion is initiated manually on a secondary computer at 3 seconds after run start. Relative to the source platform the target initially moves left 20 mm at 5 mm/s, then moves down 10 mm at 2.5 mm/s, moves right 20 mm and finally up 10 mm at a speed of 7 mm/s and stops at the original starting point. The target moves with a profile velocity with a “scurve” factor of 0 meaning the changes in velocity are linear with no curve in the trajectory. A velocity profile is shown below depicting the linear change in velocity.

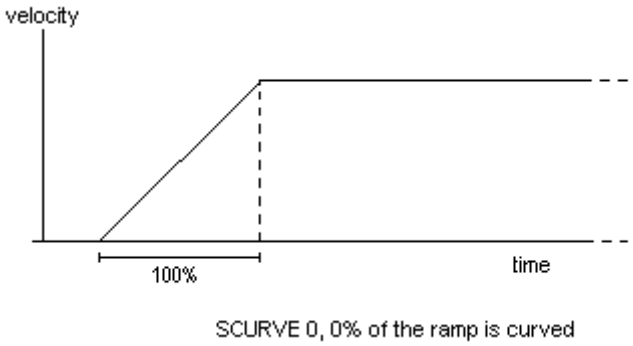


Figure 15 Target Motion Velocity Profile with “Scurve” 0 Translating to a Linear Change

The two subsystems are tested independently for performance and stability analysis using the sequences listed previously. Once completed the entire beam control system is tested with both the target motion and platform vibrations, as previously mentioned, running simultaneously for a full system experiment.

2.3 Tunnel and Isolator Effects on Jitter

The atmosphere has a major effect on the propagation of any directed energy weapon. As the beam travels downrange various disturbances in the atmosphere cause changes in the index of refraction of the propagation medium, causing the beam to refract. In order to better analyze the beam control system it is necessary to mitigate these atmospheric disturbances as much as possible. The source platform, target platform, and propagation tunnel (Figure 16) are surrounded by acrylic panels which help control atmospheric disturbances.

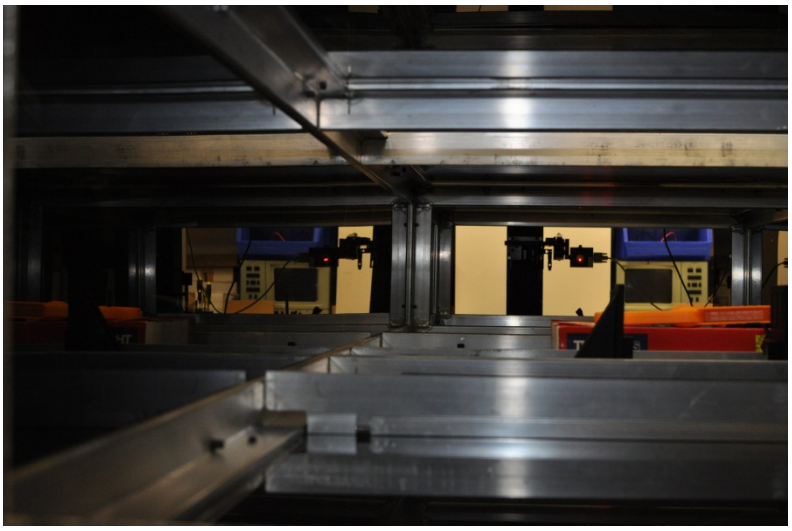


Figure 16 Enclose Propagation Tunnel Connecting Source and Target Platform

The source platform is mounted on the optical table isolators which isolate the source platform from any frequencies that would propagate through a hard-contact mount. This allows the IA's to properly induce controlled vibrations on the source platform while testing. As a baseline test the entire system was enclosed, the FSM's were locked at 0, and the optical table was floated with IA1 and IA2 turned off and the beam position was recorded.

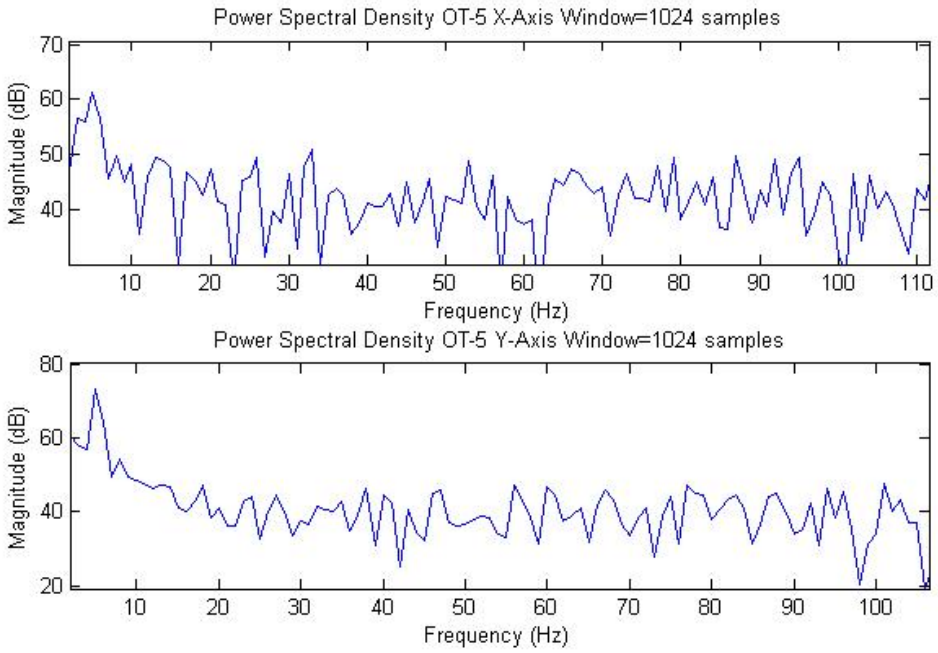


Figure 17 Tunnel Closed Power Spectral Density

Figure 17 shows the power spectral density of the beam position after the test and a 5 Hz natural frequency appears in both the x and y-direction. This natural frequency is a result of the construction of the source platform, and acoustic energy will cause the platform to vibrate at this natural frequency. Due to the sensitivity of our devices, this 5 Hz resonant frequency appears in our measurements.

3. Theory

3.1 Jitter

Jitter is defined as any deviation of the beam from its desired position, either from atmospheric turbulence or platform vibrations. Any angular deviation of the platform, or beam relative to the platform (propagation effects), of θ in an axis will cause a linear displacement of the beam at the target of *d* with relation to equation 1

$$d = r\ tan(\theta) \tag{1}$$

where *r* is the distance from the source laser to the target. A 1 µrad jitter angle at a range of 10 km will cause a deviation of 1 cm. Since jitter occurs in all axes over a period of time the end effect is a smear, which effectively increases the spot size and decreases the amount of fluence, or energy per unit area. The overall effect is shown in Figure 18. In our previous example, with a 1 µrad jitter angle at a range of 10 km, if we had a 1 cm spot size laser this would increase r from 1 cm to 3 cm, thus increasing our effective spot size by 900%.

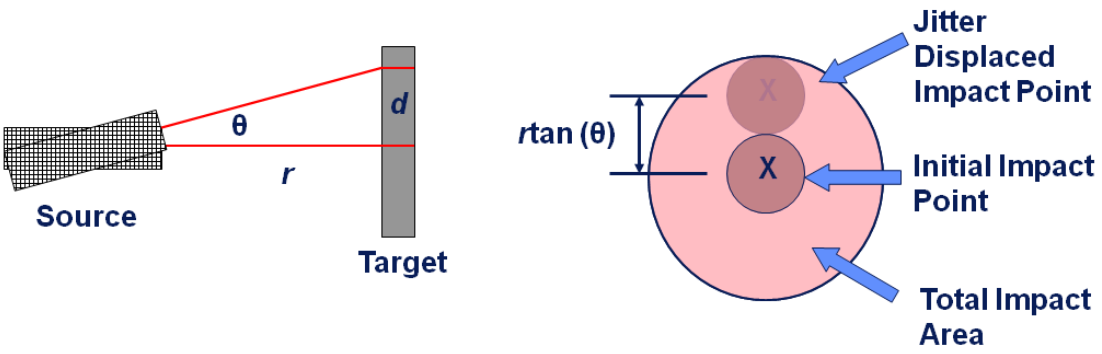


Figure 18 Jitter Displacement Diagram

3.2 Adaptive H∞ Controller

3.2.1 H∞ Control Basics

H∞ control theory allows for a controller to guarantee closed-loop stability in a system, especially in the presence of model uncertainty. In the case of this experiment, closed-loop stability with plant uncertainties is essential, since the FSM is neglected in the design process and disturbances can alter the propagation from source to target. In the FSM plant small perturbations in the model can translate to several percent of error. Considering Figure 19, stability can be guaranteed by the Small Gain Theorem, which states the product of the infinity norm$\|G\|_\infty\|H\|_\infty < 1$, where $\|\cdot\|_\infty$ is the maximum magnitude on the bode plot, for closed loop stability. *G* represents the state-space plant and *H* represents the computed controller.

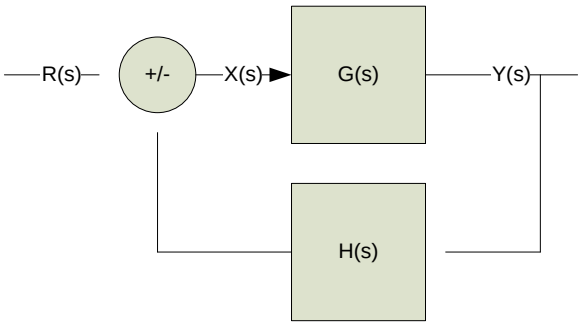


Figure 19 Sample Negative Feedback Transfer Function

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)} \tag{2}$$

The Small Gain Theorem states if $\|G\|_\infty \|H\|_\infty < 1$, then the closed-loop system is stable and this leads to the robust stabilization problem:

$$\left\| \begin{bmatrix} -K \\ 1 \end{bmatrix} (1+GK)^{-1} [1 \quad K] \right\|_\infty < \gamma \tag{3}$$

The methods of the small gain theorem can be illustrated when the plant is modeled with additive uncertainty. Figure 20 shows the closed loop system with additive uncertainty in the plant model.

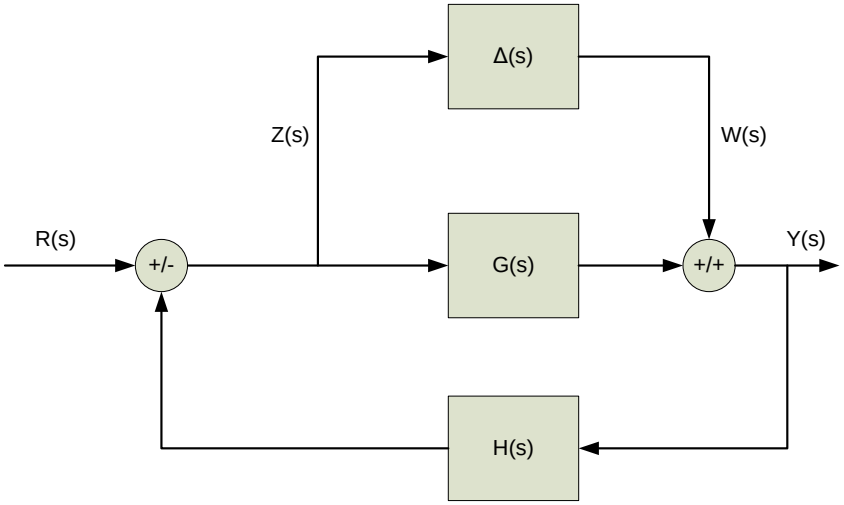


Figure 20 Sample Negative Feedback Model with Additive Uncertainty

Since our plant G(*s)* is identified, our bound for the maximum amount of uncertainty when designing the controller is contained entirely in the uncertainty term of the plant. Our system is modeled to have the uncertainty term always less than the bound of the system ε.

$$\text{Bound: } \|\Delta\|_{\infty} < \varepsilon \tag{4}$$

A normalized left co-prime factorization (NLCF) model is used to represent the plant *G(s)*. The NLCF is based on a numerator transfer function *N(s)* and a denominator transfer function *M(s)* where $G(s) = M^{-1}(s) \cdot N(s)$ as seen in equation 5.

$$G(s) = \frac{N(s)}{M(s)} \tag{5}$$

With the NLCF model our uncertainty is then modeled as small perturbations to the numerator and denominator term seen in equation 6:

$$G(s) = \frac{N(s)+\Delta_N(s)}{M(s)+\Delta_D(s)} \tag{6}$$

The NLCF model is shown in Figure 21 with the uncertainty terms as seen in equation 6

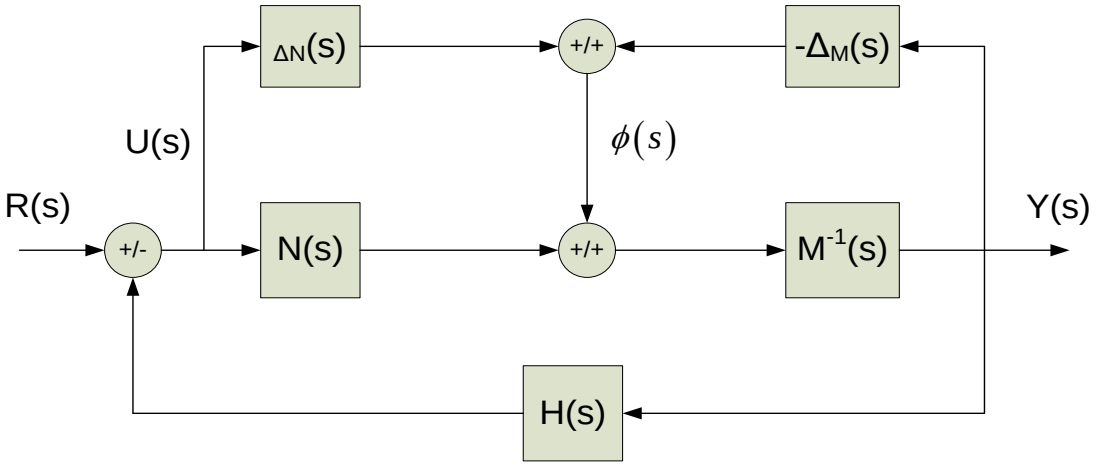


Figure 21 Normalized Left Co-prime Factorization (NLCF) Model for Additive Uncertainty

The filtering algebraic Riccati equation, equation 7, is used to calculate a unique matrix solution P, where *A, B,* and *C* are from the plant state-space model,

$$AP + PA^T - PC^TCP + BB^T = 0 \tag{7}$$

With the unique matrix solution P, we calculate the numerator and denominator polynomials *N(s)* and *M(s)*, respectively in equations 8 and 9.

$$N(s) = \begin{bmatrix} A - PC^TC & B \\ -C & 0 \end{bmatrix} \tag{8}$$

$$M(s) = \begin{bmatrix} A - PC^TC & PC^T \\ -C & 1 \end{bmatrix} \tag{9}$$

After simplifying the block diagram for the NLCF model in Figure 21 we arrive at the block diagram in Figure 22 where $T(s) = \begin{bmatrix} \frac{-H(s)}{1+G(s)H(s)} \\ \frac{1}{1+G(s)H(s)} \end{bmatrix}$.

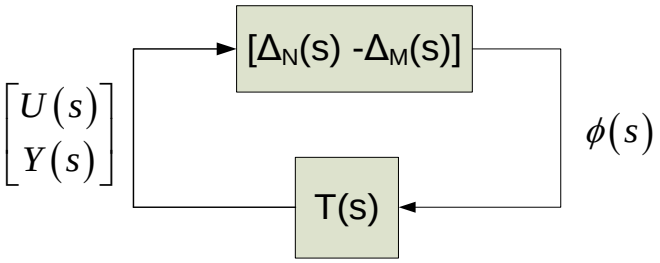


Figure 22 Simplified NLCF Block Diagram

Applying the small gain theorem, we design the controller *H(s)* of the system to never violate the theorem statement guaranteeing closed-loop stability for the system. Thus, our calculation of the necessary controller is based on the allowable uncertainty of our system. This is the basis of the small gain theorem. With our given bound, the small gain theorem becomes:

$$\|\Delta\|_\infty < \varepsilon \tag{10}$$

$$\|T\|_\infty \|\Delta\|_\infty < 1 \tag{11}$$

Therefore if $\|T\|_\infty < \frac{1}{\varepsilon}$, the closed-loop system is stable and the optimal solution to equation (3) maximizes our amount of allowable uncertainty such that

$$\|\Delta_N \quad -\Delta_D\|_\infty < \varepsilon \tag{12}$$

We now have a minimum allowable gamma $\gamma = 1/\varepsilon$ for stability. Selecting an appropriate gamma γ will lead to a successful controller. Stability, given the bounds of uncertainty with this computed controller, is guaranteed by nature of this design process.

3.2.2 Weight Design

In order to computer our desired controller K(s) there are two steps to the design process. The H∞ design process gives us one aspect of the controller design, and that is the robustness for uncertainty. The other part to the design process is designing an appropriate weight for the plant in order to achieve a desired attenuation at certain frequencies. Once an appropriate weight is chosen and amended to the plant, for our weighted plant, a minimum γ can be computed along with the stabilizing controller. In order to have an appropriate design process for the weight, a relationship between the disturbances and the output needs to be defined. Figure 23 shows a closed loop block diagram for our system with a disturbance input, D(s), which are the frequencies from the IA's.

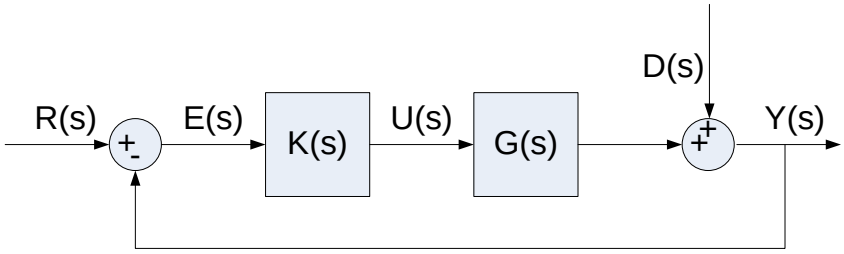


Figure 23 Closed-Loop Block Diagram for Disturbance Input

In figure 23, K(s) is our designed controller and G(s) is the plant, or FSM, and if we define L(s)=K(s)G(s) then a sensitivity function relating the propagation of the disturbance, D(s), to the output, Y(s), is calculated

$$S(s) = \frac{1}{1+L(s)} \tag{13}$$

If $|L(j\omega)|$ is large enough at frequencies of expected disturbances, the sensitivity of the output to those frequencies will be very small and the disturbance will be rejected. This gives us a foundation for our plant weighting design. The weight, W(s), needs to have high enough gains at

those disturbance frequencies in order to achieve the desired loop shape. The weights are calculated at each frequency $\omega_n = 2\pi f_n$ according the standard second ordered structure

$$W(s) = \frac{s^2+2\zeta_1\omega_n s+\omega_n^2}{s^2+2\zeta_2\omega_n s+\omega_n^2} \quad (14)$$

The weights are tuned individually for the amount of desired attenuation at each frequency by the designer adjusting the values of ζ_1 and ζ_2. As long as $\zeta_1 > \zeta_2$ then there is a positive gain at that frequency, leading to a decreased sensitivity of that frequency at the output. The ratio between ζ_1 and ζ_2 will determine the magnitude of the gain, while the individual magnitudes of ζ_1 and ζ_2 will determine the width of the weight. Below is an example for a 10 Hz tone with a 20 dB gain (ζ_1=1 ζ_2=.1).

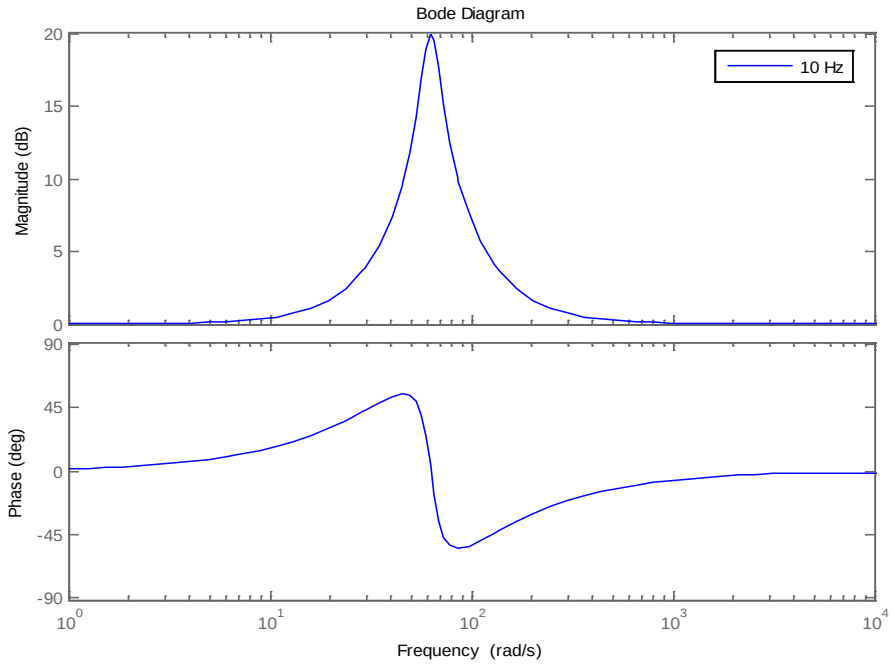


Figure 24 Bode Plot for 10 Hz (20dB Gain) using Equation 14

If a disturbance consist of multiple frequencies then individual weights can be designed then cascaded together for a final weight, $W(s)=W_1(s)\cdot W_2(s)\cdot W_3(s)$. With the combined weights W(s) can be multiplied by the plant model, G(s), to obtain a weighted plant, $G_w(s)=W(s)\cdot G(s)$, with the desired open loop shape with gains at the disturbance frequencies. Given the weighted plant with the state space realization

$$G_w(s) = \begin{bmatrix} A_f & B_f \\ C_f & D_f \end{bmatrix} = \begin{Bmatrix} \dot{\underline{x}} = A\underline{x} + B\underline{u} \\ \underline{y} = C\underline{x} + D\underline{u} \end{Bmatrix} \quad (15)$$

the robust stabilization problem, equation 3, can be solved and yield a controller $K_\infty(s)$. The solution to the robust stabilization problem involves the solutions X and Z, respectively, of the Generalized Control and Filtering Algebraic Riccati Equations (GCARE and GFARE)[10]

$$(A - BS^{-1}D^TC)^TX + X(A - BS^{-1}D^TC) - XBS^{-1}B^TX + C^T(I - DS^{-1}D^T)C = 0 \ \ (GCARE)$$

$$(A - BD^TR^{-1}C)Z + Z(A - BD^TR^{-1}C)^T - ZC^TR^{-1}CZ + B(I - D^TR^{-1}D)B^T = 0 \ \ (GFARE)$$

where $S = I + D^TD$ and $R = I + DD^T$.

Once the solutions to the GCARE and GFARE the final controller is calculated with the following realization[11].

$$K_\infty = \begin{bmatrix} A_f + B_fF + \Delta(C_f + D_fF) & -\Delta \\ B_f^TX & D_f^T \end{bmatrix}$$

$$F = -S^{-1}\left(D_f^TC_f + B_f^TX\right) \quad (16)$$

$$S = I + D_f^TD_f$$

$$\Delta = \gamma^2\left((1-\gamma^2)I - ZX\right)^{-1}ZC_f^T$$

Since the controller was calculated using the weighted plant, $G_w(s)$, the final controller needs to incorporate the final weight as well so that $K(s)=W(s)\cdot K_\infty(s)$

[10] McFarlane, D., K. Glover (1992). A Loop Shaping Design Procedure Using H∞ Synthesis. IEEE Transactions On Automatic Control, Vol. 37, no. 6, 759-769.

[11] Ibid

3.2.2.1 Bode Integral Theorem

One of the fundamental concepts in this controller design procedure is the idea of plant weighting for a desired open-loop shape with regard to the sensitivity function previously defined. As an example, let's discuss a system where a majority of unwanted disturbances range from 5 Hz to 60 Hz at random locations. The first thought might be to just use a low-pass filter that has a large gain from 5Hz to 60 Hz so the sensitivity at the output is low over that region. The problem lays within the Bode Integral Theorem which states the sensitivity of a function cannot be less than unity (0 dB) at all frequencies using output feedback with finite-bandwidth controllers. A finite-bandwidth controller is essentially any real-world device, like a FSM, that has a limit on its operating frequency. The Bode Integral Theorem, simply put, is saying that across the whole spectrum of frequencies of disturbances you cannot eliminate every single one. As you push down the sensitivity at one frequency that energy, due to the limits of a finite-bandwidth controller, needs to go somewhere, so reduction at lower frequencies will cause amplification at higher frequencies.

With this governing concept it can be seen why certain frequencies need to be targeted. That is the reason why the weights take the form of equation 14, and can be tuned using the ζ terms, so that specific frequencies are attenuated and the spillover to higher frequencies is minimized.

3.2.3 Assumptions and Real-time Realization

The feasibility of our adaptive H_∞ controller rests on the ability to calculate the controller in real-time. The GCARE and GFARE's are the time demanding calculations and the limiting factor in the real-time calculation. Our current system is a serial programming environment, not allowing for multi-threaded programming, like calculating the controller in a separate thread and updating when it was complete. Given the assumption that the primary frequencies of unwanted jitter are well below the bandwidth of our mirror, the FSM model can be neglected (G(s) = 1). Our FSM is modeled as a 2nd ordered system

$$\frac{\theta(s)}{\theta_d(s)} = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$
$$\frac{Y(s)}{\theta_d(s)} = \frac{2r\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \qquad (17)$$

where ω_n = 1300 rad/sec, r = 4.9 m, and ζ = 0.591. $\theta(s)$ is the actual FSM angle, and $\theta_d(s)$ is the desired angle, and Y(s) is the actual beam position on the target. Since the angles used on the FSM are relatively small, microradian, small angle approximation can be used which states $d=2r\theta$. The poles are large enough that the settling time for smaller frequencies is negligible and our assumption is valid.

With our assumption that the plant G(s) = 1, our weighted plant model just becomes our designed weight. The parameterized solution to the GFARE is computed with an observer canonical realization of the weight[12]

$$W(s) = \left[\begin{bmatrix} -2\zeta_2\omega & 1 \\ -\omega^2 & 0 \end{bmatrix} \quad \begin{bmatrix} 2\omega(\zeta_1 - \zeta_2) \\ 0 \end{bmatrix} \right] = \begin{bmatrix} A_f & B_f \\ C_f & D_f \end{bmatrix} \tag{18}$$

$$[1 \quad 0] \qquad\qquad 1$$

where ω is the identified frequency. With the given weight model and plant assumption the solution to the GFARE becomes[13]

$$Z = \begin{bmatrix} z_1 & 0 \\ 0 & \omega^2 z_1 \end{bmatrix}$$

$$z_1 = 2\omega\left(\sqrt{(\zeta_1+\zeta_2)^2 + (\zeta_1-\zeta_2)^2} - (\zeta_1+\zeta_2)\right) \tag{19}$$

The parameterized solution for the GCARE is solved first using a controller canonical realization of the weight[14]

$$W(s) = \left[\begin{bmatrix} -2\zeta_2\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right] \tag{20}$$

$$[2\omega(\zeta_1 - \zeta_2) \quad 0] \qquad 1$$

Using the realization in equation 20 the solution to the GCARE is identical to equation 19[15]

$$X_c = \begin{bmatrix} x_{c_1} & 0 \\ 0 & \omega^2 x_{c_1} \end{bmatrix}$$

$$X_{c_1} = 2\omega\left(\sqrt{(\zeta_1+\zeta_2)^2 + (\zeta_1-\zeta_2)^2} - (\zeta_1+\zeta_2)\right) \tag{21}$$

---

[12] O'Brien, R.T; Watkins, R.J; (2011) Adaptive H_∞ vibration control., Dynamic Systems and Control Conference 2011
[13] *Ibid*
[14] *Ibid*
[15] *Ibid*

In order to make the solution to the GCARE in equation 21 compatible with the observer canonical form of the weight in equation 18, a transformation matrix from controller canonical (equation 20) to observer canonical form (equation 18) is given by

$$T = \begin{bmatrix} 2\omega(\zeta_1 - \zeta_2) & 0 \\ 0 & -2\omega^3(\zeta_1 - \zeta_2) \end{bmatrix} \tag{22}$$

The transformation matrix can then be used to convert the solution to GCARE in equation 21 to be compatible with the weight realization in equation 18.

$$X = T^{-T} X_c T^{-1} = \begin{bmatrix} x_1 & 0 \\ 0 & \frac{x_1}{\omega^2} \end{bmatrix}$$[16]

$$x_1 = \frac{\sqrt{(\zeta_1+\zeta_2)^2+(\zeta_1-\zeta_2)^2}-(\zeta_1+\zeta_2)}{2\omega(\zeta_1-\zeta_2)^2} \tag{23}$$

Now that we have parameterized solutions for GCARE and GFARE we can establish an adaptive H_∞ design process. The elegance of this controller is both the ability to design with direct frequency input and to maintain simple calculations by computing individual controllers and weights for each frequency, then cascading them together by multiplication as you would for transfer functions in series.[17]

Step 1: First choose a baseline weight, $W_0(s)$, and stabilizing H_∞ controller, $K_0(s)$.

In our case a 1/2r term is used to cancel broadband 20 dB gain that is caused by the 2r term in equation 17, from the small angle approximation.

Step 2: Next, identify a set number (N) of frequencies $(f_1, \dots, f_N)$ based on the feedback from the output sensor (OT-5 in this experiment).

Step 3: For each frequency, f_i, select the dampening ratios $\zeta_1^i \; and \; \zeta_2^i$, for the attenuation weights $W_{f_i}(s)$ from equation 14 and compute the H_∞ controller $K_{f_i}(s)$ from equation 16, using the solutions of the GCARE and GFARE in equations 19 and 23 respectively.

---

[16] O'Brien, R.T; Watkins, R.J; (2011) Adaptive H_∞ vibration control., Dynamic Systems and Control Conference 2011

[17] *Ibid*

Step 4: Compute the final controller by cascading the individual controllers and weights together from step 3 with the following form

$$K(s) = K_0(s)W_o(s)\left[\prod_{i=1}^{N} K_{f_i}(s)W_{f_i}(s)\right] \qquad (24)$$

The actual number of frequencies in step 2 has a range of possibilities and is all dependent upon the operating platform and desired performance by the individual. In step 3, when the dampening ratios are chosen, it is important to note that the ratio between the two determines the gain at that frequency, but the magnitude of the two numbers will determine how wide the actual gain is in the Bode plot. If the numbers are large then the range of the gain will be wider, and narrower if there magnitudes are smaller, as seen in Figure 25 where the gain is 20 dB at 10 Hz but the width is wider for$\zeta_1 = 10\ and\ \zeta_2 = 1$.

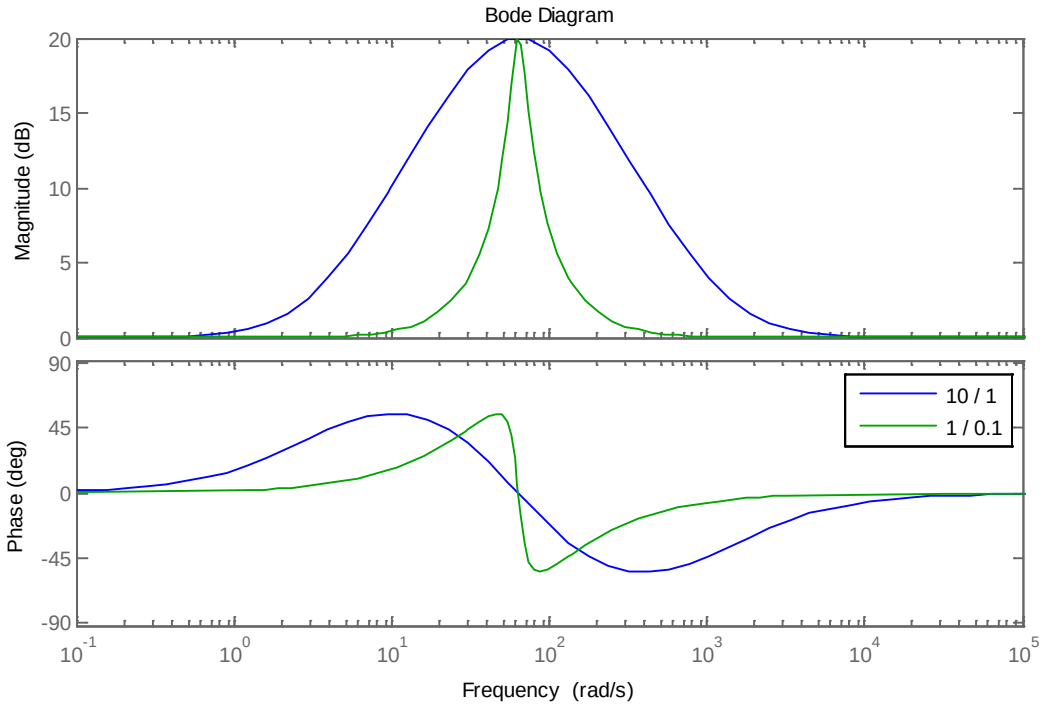


Figure 25 Bode Plot of Weight in Equation 14 Form with Same Ratio, Varying Magnitudes of damping coefficients

This leads to a situation where the dampening ratios need to be selected so that they are wide enough to cover a range for the frequency identified but no so large there is attenuation across a

wide range of frequencies. The issue arises, when combining the controllers in step 4, is that if you have two frequencies relatively close (~5 Hz or even more) then the multiple controllers will contribute to the gain of that frequency, causing more attenuation then desired. This can lead to spillover at higher frequencies. While it seems reasonable to use a smaller ζ, the resulting system performance is less stable as poles of the system come closer to the imaginary axis. To obtain the desired amount of attenuation at each frequency, Step 3 becomes an optimization problem.

3.2.3.1 Damping Ratio Optimization

As mentioned previously the problem with the adaptive design process is a tradeoff between stability and ease of calculation. To avoid optimization, smaller magnitude damping ratios can be used but performance may suffer. With optimization there is better performance, but it requires more calculation time. A simple non-linear optimization technique is the Nelder-Mead simplex approach, which is a heuristic search method for local minima. The Nelder-Mead method essentially creates a multi-dimensional polyhedron, with each vertex constructed from possible values for the unknown variables in the equation to be minimized. The value of the equation is calculated at each vertex and the largest value is reflected across the center of gravity and tested. The best way to visualize this is a triangle on an elevation map flipping over itself down the hill to the local bottom in the valley.

The first step is to create an error function, which relates all the damping ratios, which can be minimized. The best way is to create an error function that is the sum of squared errors (SSE) between each desired attenuation level and the actual attenuation level from the final controller in equation 24

$$Error = \sum_{i=1}^{N}(|L(j\omega_i)|_{dB} - |L_{desired}(j\omega_i)|_{dB}) \tag{25}$$

The loop gain at each identified frequency, $\omega_i = 2\pi f_i, i = 1, \dots, n$, is computed from

$$|L(i\omega_i)|_{dB} = |L_0(j\omega_i)|_{dB} \prod_{l=1}^{N} \left|K_{f_l}(j\omega_i)\right|_{dB} \left|W_{f_l}(j\omega_i)\right|_{dB} \tag{26}$$

where $L_o(s) = G(s)K_0(s)K_0(s)$, from Step 1 of the adaptive design process and the product function is calculating the gain at ω_i contributed from each individual weight and controller. A

lookup table for values of $|L_0(j\omega_i)|_{dB}$ can be pre-calculated since it is not dependent on the damping ratio. The terms $W_{f_l} and\ K_{f_l}$ are the weights for each of the identified frequencies f_l and can be computed as a function of $\zeta_1^l and\ \zeta_2^l, l = 1, \dots, n$, as follows

$$\left|W_{f_l}(j\omega_i)\right|_{dB} = \begin{cases} -20\log_{10}(\zeta_2^l) + 20\log_{10}(\zeta_1^l) & \omega_i = \omega_l \\ 20\log_{10}\sqrt{\dfrac{(\omega_i-\omega_l)^2 + 4(\zeta_1^l)^2\omega_i^2\omega_l^2}{(\omega_i-\omega_l)^2 + 4(\zeta_1^l)^2\omega_i^2\omega_l^2}} & \omega_i \neq \omega_l \end{cases}$$

$$\left|K_{f_l}(j\omega_i)\right|_{dB} = 10\log_{10}\left(\left(\boldsymbol{R}\left(K_{f_l}(j\omega_i)\right)\right)^2 + \left(\boldsymbol{I}\left(K_{f_l}(j\omega_i)\right)\right)^2\right)$$

Where **R** and **I** indicate the real and imaginary components of a complex number.

$$\boldsymbol{R}\left(K_{f_l}(j\omega_i)\right) = 1 + \frac{2\alpha^2\gamma_l^2\delta a_{11}\omega_i^2\omega_l}{\left(\left((\gamma_l^2-1)\delta^2 - \alpha^2\right)((\omega_i^2-\omega_l^2)^2 + a_{11}^2\omega_i^2)\right)}$$

$$\boldsymbol{I}\left(K_{f_l}(j\omega_i)\right) = 1 + \frac{2\alpha^2\gamma_l^2\delta\omega_i\omega_l(\omega_i^2-\omega_l^2)}{\left(\left((\gamma_l^2-1)\delta^2 - \alpha^2\right)((\omega_i^2-\omega_l^2)^2 + a_{11}^2\omega_i^2)\right)}$$

$$\delta = \zeta_1^l - \zeta_2^l, \sigma = (\zeta_1^l + \zeta_2^l, \alpha = \sqrt{\delta^2+\sigma^2} - \sigma$$

$$a_{11} = \frac{\omega_l\left(2\zeta_2^l + \alpha + \delta + \gamma_l^2\alpha\delta(\delta-\alpha)\right)}{(\gamma_l^2-1)\delta^2 - \alpha^2}$$

With a parameterized solution to solve for the error function in equation 25, with respect to the damping ratios, a minimization method can be chosen and implemented. When testing with the possibilities of optimization in the experiment, the Nelder-Mead simplex approach was chosen due to its ability to handle multiple variables and a non-linear function, and the ease of programming the algorithm. The value of $\zeta_2^l, l = 1, \dots, n$, was set to 0.1 so that only the top damping ratios had to be optimized, and it is known that a magnitude of 0.1 will yield a stable controller for our system with the corresponding dampening ratio, ζ_2^l.

Nelder-Mead is only one example of an optimization method that can be used in parallel with equation 25.

3.3 Proportional-Integral (PI) Control

3.3.1 PI Control Theory

A proportional-integral (PI) controller is used to control FSM B for the target tracking, as well as FSM A as a comparison for jitter control. A PI controller seeks to minimize the error between the actual and desired value of a system variable. The proportional component produces a control signal that is proportional to the error signal and instantly reacts to a change in error signal. The integral component produces a control signal that is proportional to the integral of the error of the run time. It accumulates the instantaneous error over time and accounts for error that was not corrected for previously. The integral term is responsible for removing much of the steady-state error in a system.

3.3.2 PI Controller Gains

While H_∞ allows for some uncertainty in a model, direct tuning for a PI controller requires a fairly accurate plant model. All though we have a model for the FSM's they still have some uncertainties so a testing based tuning method is needed. One method of PI tuning, the Zeigler-Nichols method, does not require a plant model and is a heuristic search method. First the integral and derivative terms are set to 0 and the proportional term is increased until the system output begins to oscillate. The proportional gain at which the system operates then becomes the ultimate gain, K_u, and the period of the oscillation is T_u. K_u and T_u are then used to calculate the optimal gains, depending on the controller type, with the following Zeigler-Nichols rules

Controller	K_p	K_i	K_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$1.2K_p/T_u$	-
PID	$0.6K_u$	$2K_p/T_u$	$K_pT_u/8$

Table 1 Zeigler-Nichols Tuning Rules for PID Based on Critical Gain and Critical Period

As seen in previous experiments these gains provided an unstable response in our system, but provided a baseline. The values were reduced by 50% then manually tuned for each scenario. Table 2 provides the PI gains for FSM A when being used to compare against H_∞.

FSM Axis	Gain	Value
X	Proportional	0.018

	Integral	21.6
Y	Proportional	0.0225
	Integral	27.0

Table 2 FSM A PI Gains for Jitter Control Used for Comparison Against H_∞

The target tracking controller was designed using the same methods and required manual tuning as well, however in section 2.2.1 it was stated the target tracking controller utilizes a second order integral term. This is done by putting two PI controllers in series with each other, for each axis. This is needed to deal with a piecewise velocity profile. When the velocity starts, the position plot will be a line with constant slope, and the first PI controller will take care of the initial change in velocity and follow that slope. The actual versus desired position will have parallel lines but there will be a steady-state error, and the integral term is already handling the velocity portion of the error. The second PI controller is what will remove that "bias" caused by the ramp in the position profile.

FSM Axis	Gain	Value
X	Proportional 1st	0.044
	Integral 1st	0.81
	Proportional 2nd	0.26
	Integral 2nd	4.6
Y	Proportional 1st	0.044
	Integral 1st	0.81
	Proportional 2nd	0.26
	Integral 2nd	4.6

Table 3 FSM B PI Gains for Target Tracking

3.4 Frequency Identification

The ability of our system to actively control the jitter experienced on the platform directly relies on the ability to accurately identify the tonal frequencies of vibration. The Fast Fourier Transform (FFT) is a well-recognized and standard method for computing the discrete Fourier Transform (DFT). The DFT is a frequency-domain representation of a signal which allows direct analysis of a signal to find the component frequencies within. Two important elements concerned with an FFT are the number of samples (N) and the sample period (T) of the system.

Our system is already defined to operate at a sample rate of 1 kHz, so our period T=0.001s.. The range of frequencies we can identify is also determined by the sample rate, and is

2/T = 500 Hz is the max frequency that can be identified. The last characteristic of the FFT is the resolution, which is 1/(NT), where N is the number of total points used. We discussed that our weights have a small range that they affect, +/- 2 Hz, so a resolution of 1 Hz is sufficient. Choosing a value for N of 1024 gives us slightly better than 1 Hz resolution, but is also a power of 2 which makes the FFT calculation much more efficient, and no need for zero-padding.

3.5 SIMULINK Model

The experiments conducted in the USNA Directed Energy Beam Control Laboratory are designed and run using the SIMULINK environment in MATLAB. SIMULINK allows for block based models to be created, allowing users to connect sensors and hardware, and then be compiled into a C environment and ran in real-time. The SIMULINK model for the beam control system is shown in Figure 26.

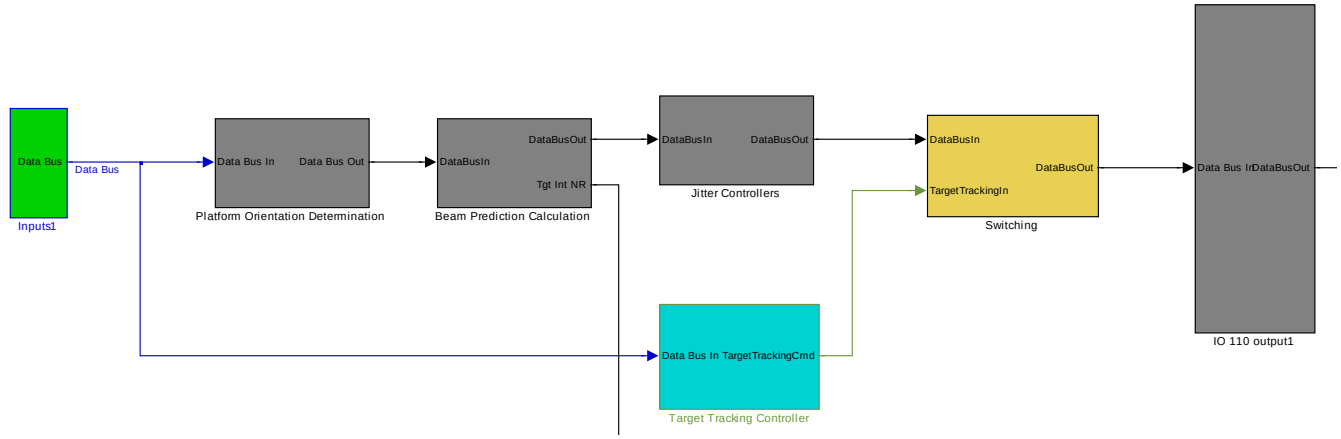


Figure 26 Beam Control System SIMULINK Model

Every component and operation of the beam control system required to run the experiment is defined in a different block. The green block receives all the input signals from the sensors and equipment in the system. The grey block to the far right sends all the command signals and outputs all the signals and stored data back to MATLAB. The first two grey blocks to the right of the green block, are beam position calculators used in other experiments but left for future implementation in a feed-forward based design. The middle grey block contains the both the PI and H_∞ controllers for jitter mitigation. The blue block contains the PI controller for target tracking. The yellow block allows the user to select and switch between different controller and setups, for example between the PI or H_∞ jitter controller.

Within the Jitter control block are two subsystems shown in Figure 27 which contain the two separate jitter controllers.

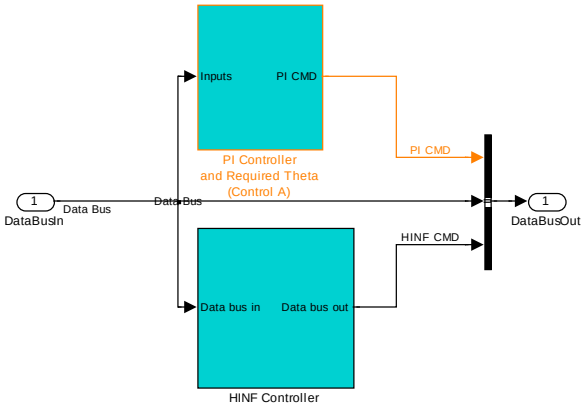


Figure 27 Jitter Control Subsystem Block

The top block is the PI controller and the bottom block is the H_∞ controller. Figure 28 shows the PI controller subsystem and Figure 29 shows the H_∞ controller subsystem.

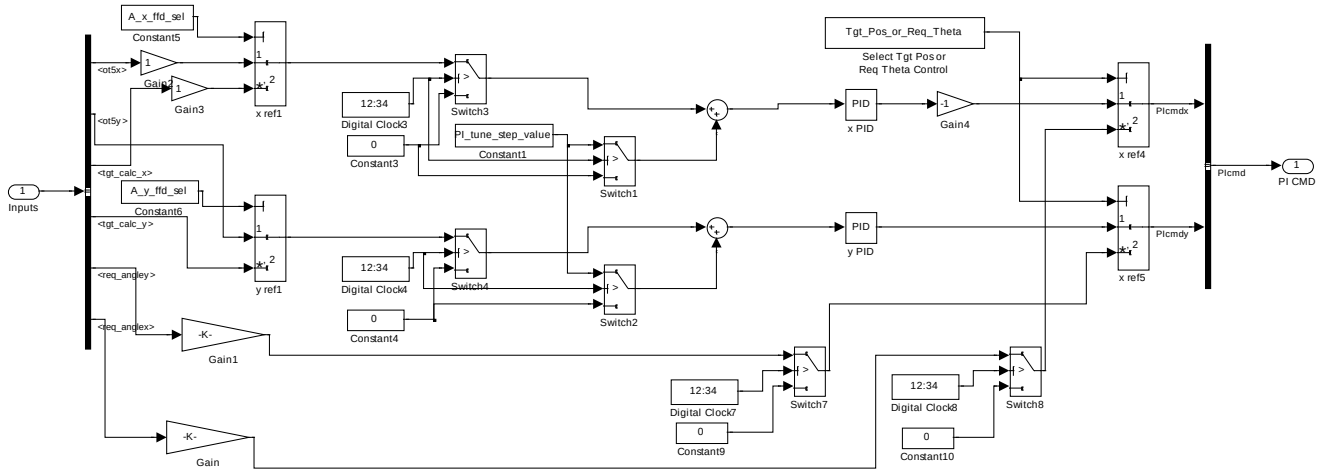


Figure 28 PI Jitter Controller SIMULINK Model

The different paths allow for different sensor feedback to be used in the jitter control, however only the top two paths are used in this experiment. The first two outputs of the demux block are the x and y position read out from OT5 that are fed through switches and then into

independent PI controllers. The switches allow the user to select which controller is on, and when to engage it during the experiment.

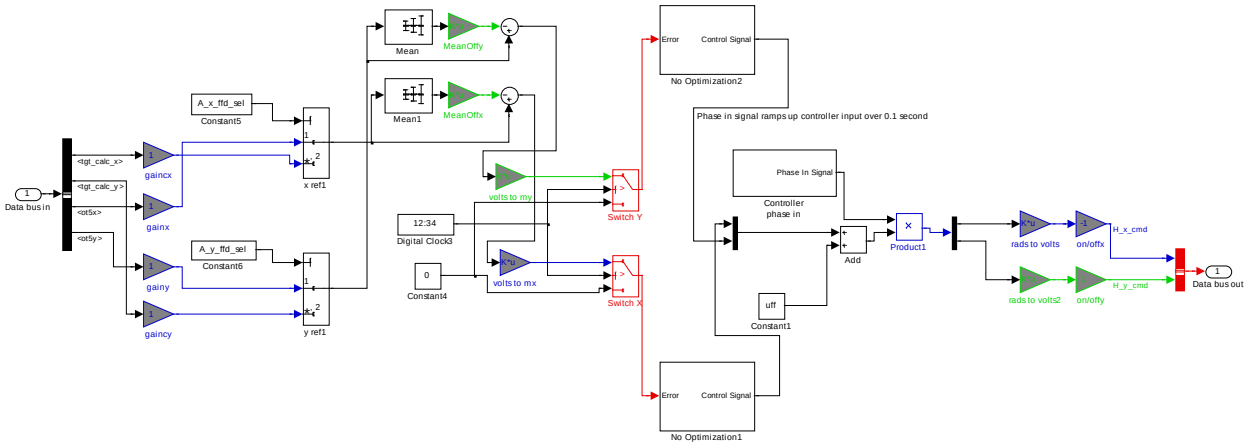


Figure 29 H_∞ Controller Subsystem

Similar to the PI controller, two different error signals are fed into the system, but OT5 is used in this experiment. The x and y positions are fed into individual H_∞ controllers and then into gain blocks which convert a desired FSM angle into a corresponding voltage signal for each axis. A phase in block is used so that when the controller first turns out, there is not an instantaneous shock-command signal to the system, causing instability. Figure 30 is the subsystem of the individual H_∞ controllers.

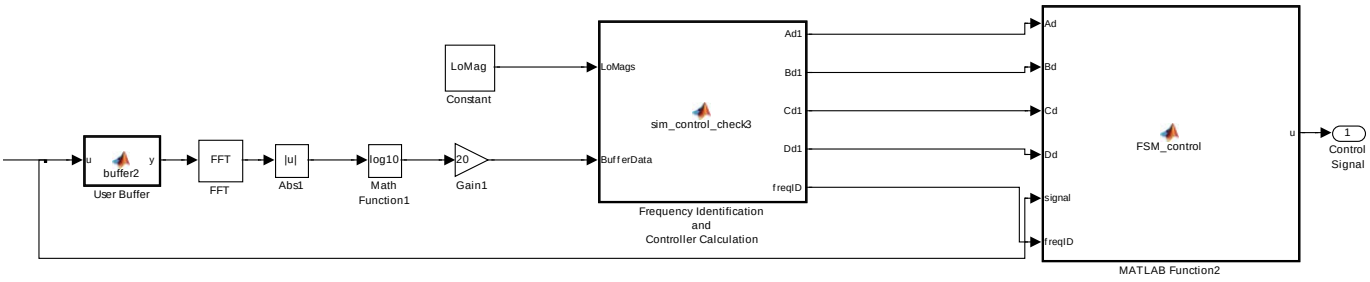


Figure 30 H_∞ Controller Block SIMULINK Model

The blocks in this system follow the steps set forth in section 3.2.3 for the adaptive design process. The first block is a buffer block which stores the 1024 samples necessary to perform the frequency identification. The next four blocks are an FFT function which take the

buffer data samples and feed out the frequencies and their respective magnitudes of disturbance. The next larger block, takes the FFT data and performs an identification of the largest frequencies of disturbance, then uses them to calculate the corresponding controller. That block outputs the corresponding state space control matrices (A,B,C,and D) along with the identified frequencies. The last block then takes in the outputs of the previous block and first checks if the newly identified frequencies differ from those used for the current controller. If no significant change has occurred the controller is not changed, otherwise the new state-space control matrices are updated and converted to a discretized model. While the frequency identification and controller update only occur every 1024 time steps, the last block inputs the error signal and uses it to update the controller and produce the control signal every time step in parallel with the other calculations.

4. Experimental Results

4.1 Jitter Controller Performance

4.1.1 Statistics Used

The following statistics were used when evaluating the jitter mitigation portion of the beam control system:

- x and y position measurement of beam position on target
- power spectral density of beam position on target in x and y direction
- jitter angle
- root mean square (RMS) of the jitter angle
- percent improvement of RMS jitter angle
- standard deviation of jitter angle
- percent improvement of standard deviation of jitter angle

The x and y position measurements are good initial indicators for the magnitude of jitter reduction as well as the accuracy of the system, which can be observed visually.

The power spectral density of the beam position in both the x and y position was determined using Welch's method in MATLAB with a window sample of 1024 and an overlap of 50%. The output of the power spectral density is a measurement of the power (dB/Hz) experienced at each frequency, and gives a full spectrum view of the various frequencies of disturbance in the output. It provides a means of directly measuring the effect of the jitter controller in seeing which frequencies it detects and how much it attenuates at the respective frequencies.

Jitter angle is the angular miss distance of the beam at the target with respect to the platform. It is calculated for each sample, by means of small angle approximation, by dividing the radial miss distance at the target by the distance between the target and source platform. The jitter angle provides a means of normalizing the data, independent of target range, so it can be compared to other systems.

The RMS of the jitter angle is a good measurement of the accuracy of the beam control system. The RMS of the jitter angle also indicates the RMS radial miss distance from the

calibrated center of the target. The RMS jitter angle is computed during the entire range of the given disturbance set of frequencies, and when being controlled it is calculated through the entire length of time during which the controller is active.

The percent improvement of the RMS jitter angle indicates the improvement between the uncontrolled set of frequencies and when the controller is actually engaged.

The standard deviation of the jitter angle is an indicator of the precision of the beam control system.

The percent improvement of the standard deviation measures the improvement of the beam control system precision once the controller is engaged as compared to no controller.

4.1.2 Jitter Mitigation for 12Hz and 38Hz shifted to 9Hz and 46Hz disturbance

The jitter control subsystem is tested independent of the entire beam control subsystem by using a stationary target and a fixed FSM B. The experiment uses two sets of two frequencies that shift midway through the run: 12 Hz and 38 Hz initially and then 9 Hz and 46 Hz at 7 seconds after run start. Two controllers, PI and adaptive H_∞, are tested and compared against each other for overall improvement with respect to an uncontrolled jitter run. Both controllers utilize error feedback from the target sensor, OT5, and are computationally, unaware of the input disturbance frequencies.

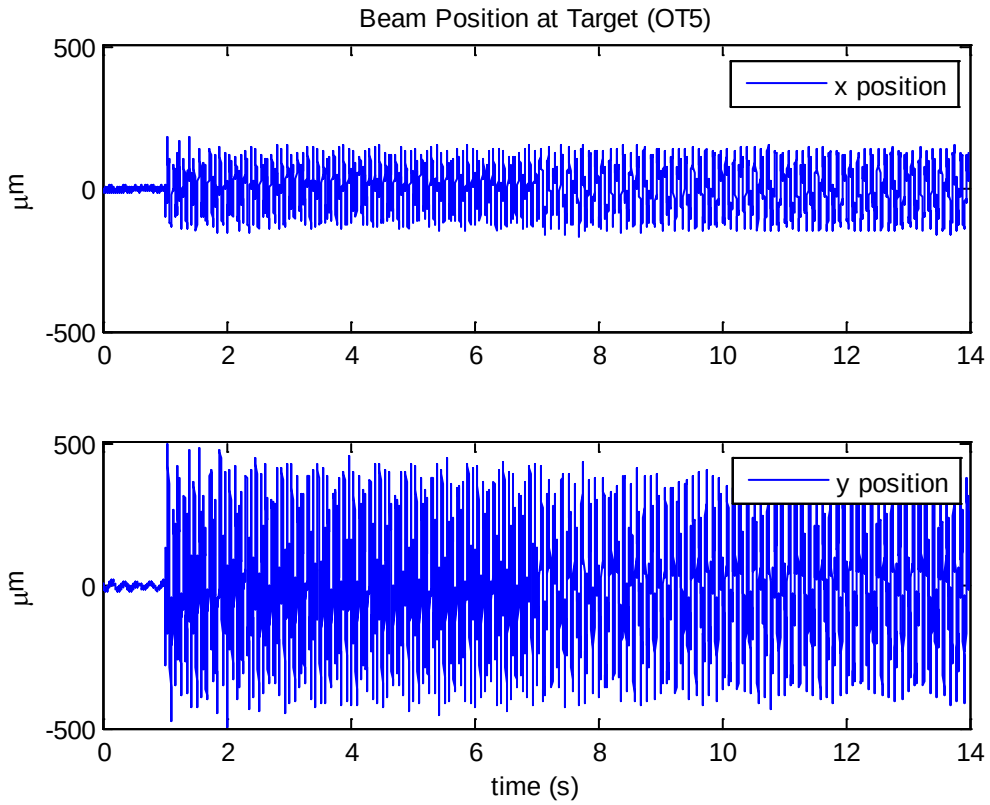


Figure 31 Beam Position for Uncontrolled Run with Multiple Changing Frequencies

Figure 31 shows the uncontrolled run with the two sets of frequencies that will be used throughout the experiment. The timing and magnitude of the disturbance will remain the same in every test. The first second of the run there is no input disturbance, then from 2 to 7 seconds is the first set of frequencies, and at 7 seconds the second set of frequencies are switched in. This uncontrolled run will be our basis for comparison between the two controllers for the amount of improvement.

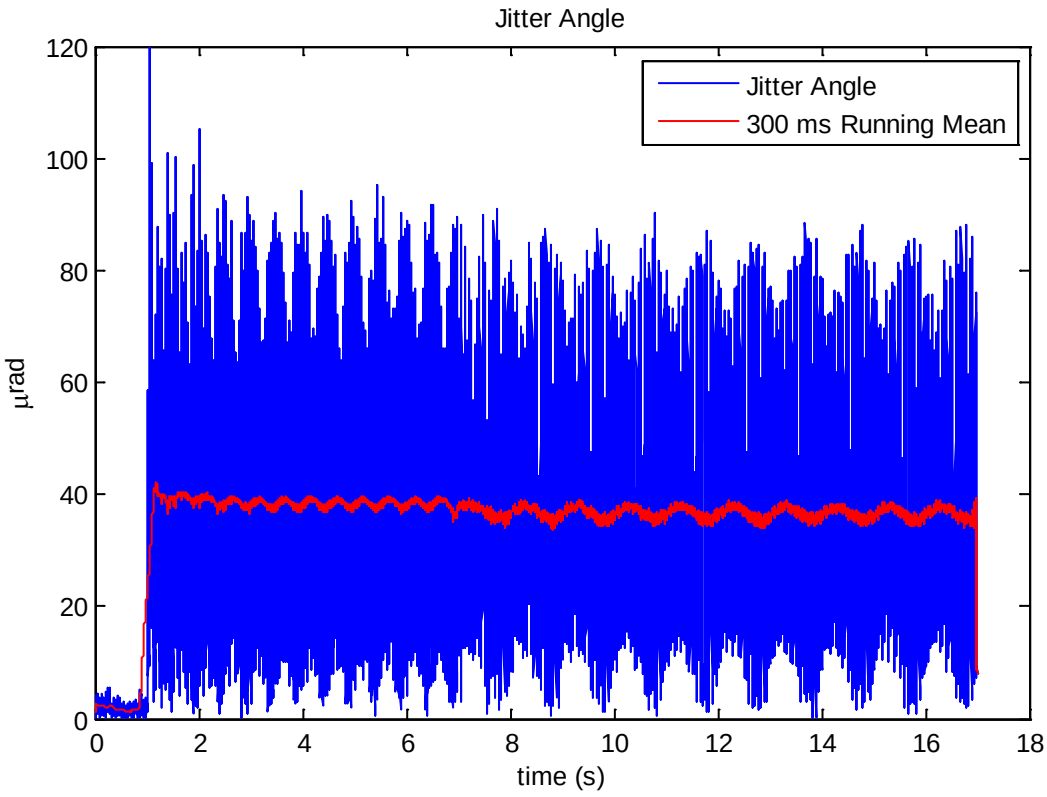


Figure 32 Jitter Angle for Uncontrolled Run with 300 ms Running Mean

Figure 32 shows the jitter angle of the uncontrolled system. The jitter angle allows the system to be measured on a normalized scale, independent of length, so it can be readily compared to other systems. The uncontrolled jitter angles oscillated between 0 µrad and 85 µrad.

The uncontrolled beam had a RMS of 42.2 µrad and a standard deviation of 22.2 µrad. These two measurements will be the base measurements for comparing the PI and H_∞ controllers.

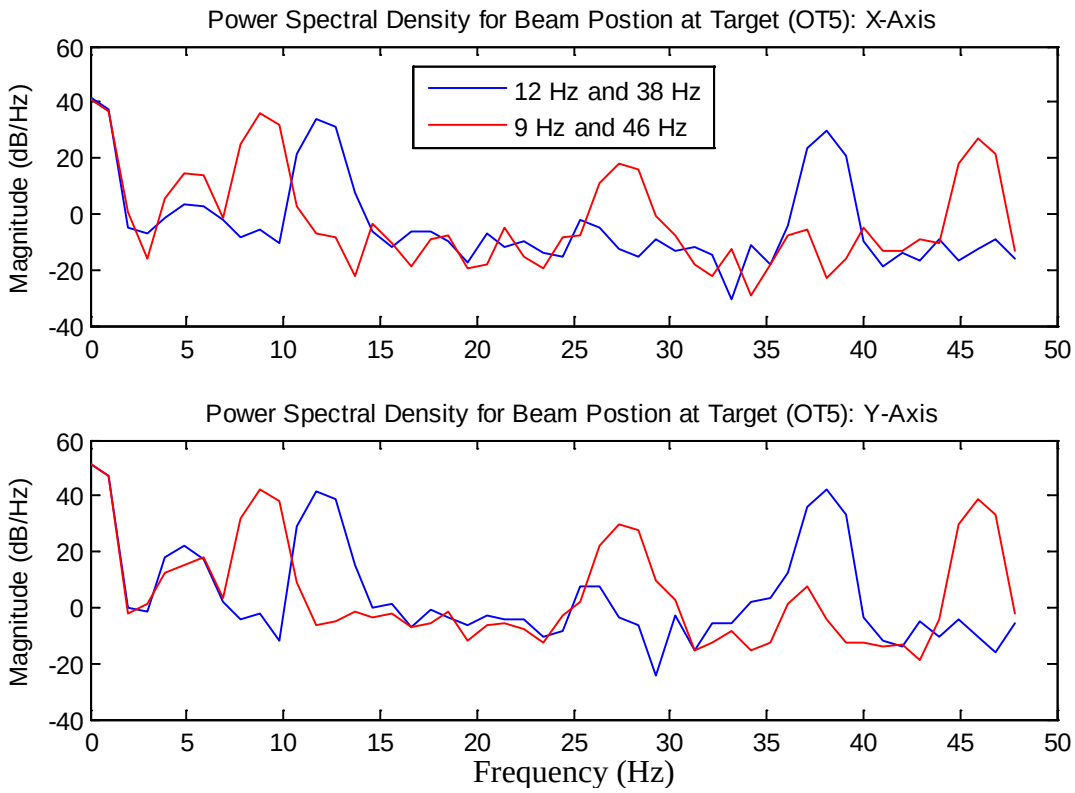


Figure 33 Power Spectral Density of Uncontrolled Beam Baseline Run

Figure 33 shows the power spectral density of the target data that were created using 1024 samples from the time period of the given disturbances. The tonal frequencies are apparent, along with the 5 Hz natural frequency of the platform and a tone at 27 Hz which is a resonance of the 9 Hz signal.

4.1.2.1 PI Jitter Control

The PI jitter controller initiated at 3.5 seconds after run start, allowing the disturbance frequencies to reach steady state. The controller remained on through the entire run, with the disturbance shifting at 7 seconds.

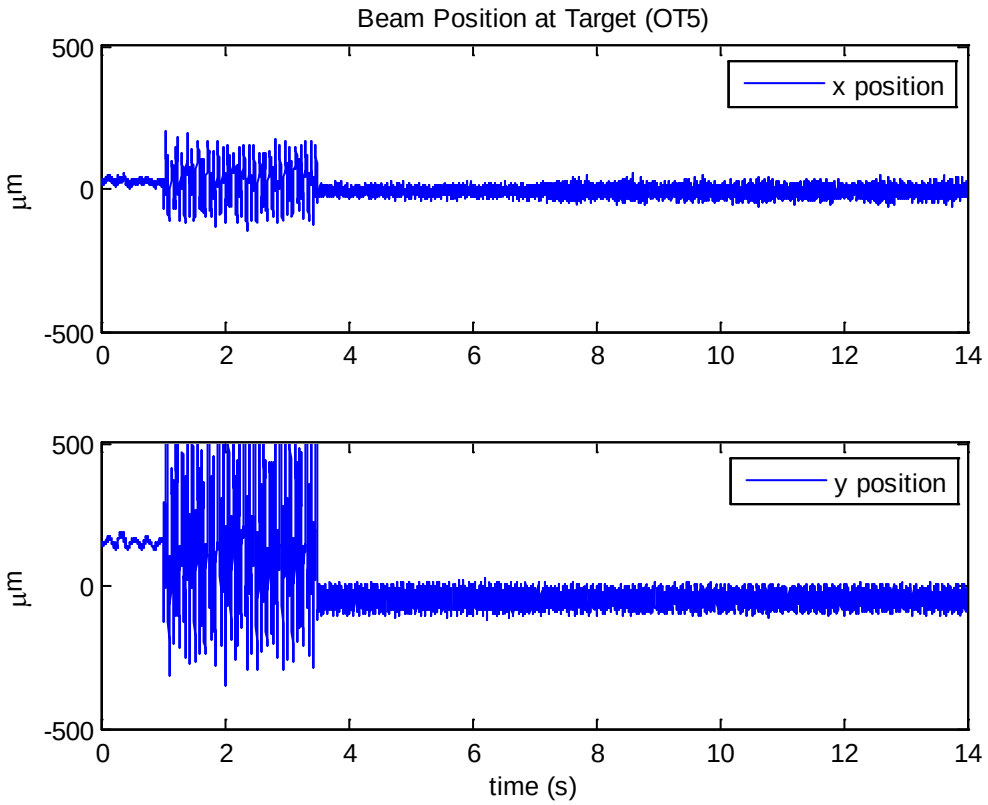


Figure 34 Beam Position With PI Jitter Controller and Multiple Shifting Frequencies

Figure 34 shows the position of the beam on the target with the PI jitter controller. With initial visual inspection, it is apparent that the controller had a significant impact on reducing the jitter and providing a tighter beam on the target. Also in the x-axis, when looking at the 7 second mark, it is visible where the frequencies change and the attenuation degrades.

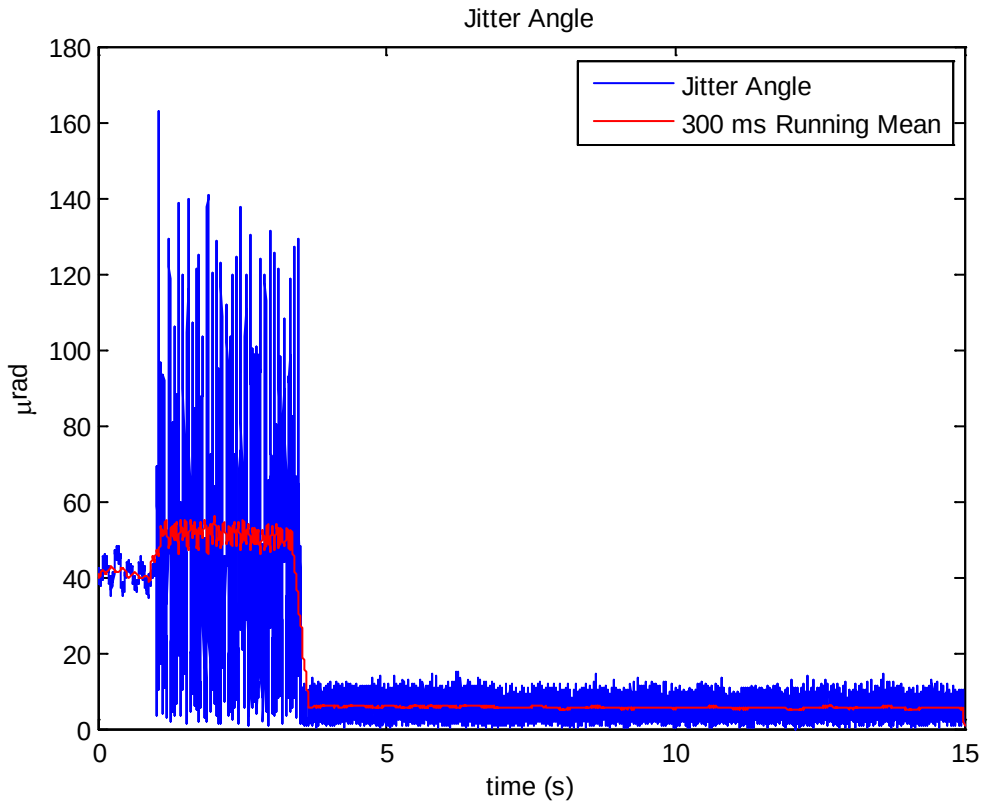


Figure 35 Jitter Angle and 300 ms Running Mean for PI controller with Stationary Target

Figure 35 shows the jitter angle for the PI controller along with the 300 ms running mean. The RMS jitter angle for the controlled beam is reduced 85.4% to 6.2 µrad. The standard deviation of the jitter angle is reduced 88.3% to 2.6 µrad.

4.1.2.2 Adaptive H_∞ Jitter Controller

This test used the same disturbance setup as in the PI controller setup except now with the adaptive H_∞ controller. The H_∞ controller is initiated at 2.048 seconds after the start of the run.

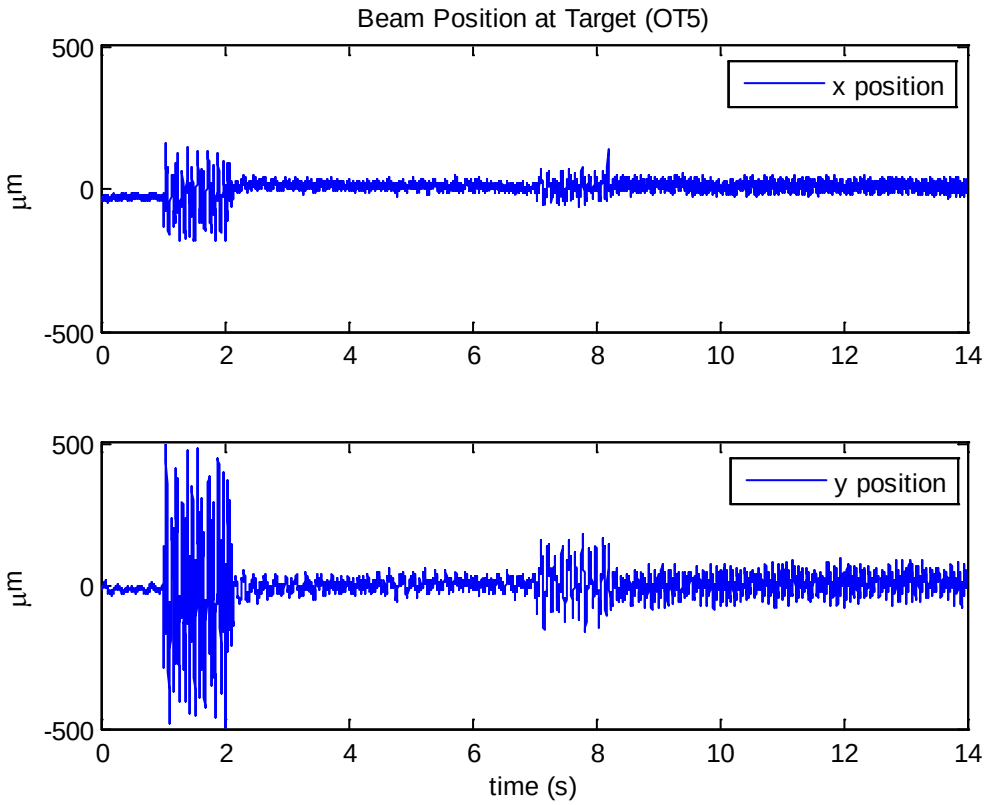


Figure 36 Beam Position With Adaptive H∞ Controller and Multiple Shifting Frequencies

Figure 36 shows the position of the beam at the target with the adaptive H_∞ controller engaged at 2.048 seconds after the run start. Similar to the PI controller, the Adaptive H_∞ controller shows significant improvement on the position plot. The change in frequencies is more noticeable than in the PI controller due to the frequency specific tuning of the H_∞ control algorithm.

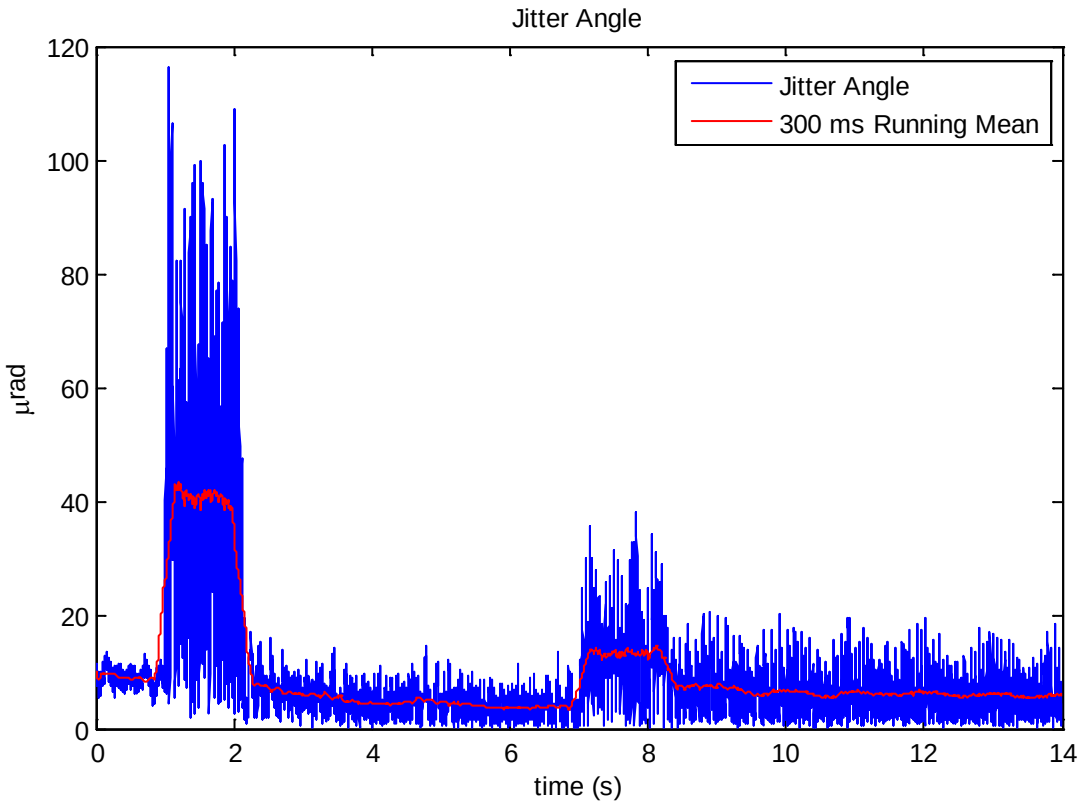


Figure 37 Jitter Angle for Adaptive H∞ Jitter Controller with Multiple

Figure 37 shows the jitter angle of the adaptive H_∞ controller along with a 300 ms running mean. The RMS of the jitter angle is reduced 80% to 8.1 µrad. The standard deviation is reduced 78.4% to 4.8 µrad. The adaptive H_∞ controller is successful at identifying and attenuating the tonal frequencies, however when compared to the RMS and standard deviation of the PI jitter controller it performs slightly less well. A main source of the underperformance is the 7 second to 8 second time frame when the controller needs to identify the new frequencies.

4.1.2.3 Controller Performance Comparison

Table 4 shows the comparison of the various controller performance variables. The PI controller provided better performance, however when looking at the standard deviation and RMS of the adaptive H_∞ controller, when it is tuned to the disturbances, it provides better performance than the PI controller.

Control Scheme	RMS Jitter Angle (μrad)	Improvement in RMS Jitter Angle	Standard Deviation (μrad)	Improvement in Standard Deviation
Uncontrolled	42.2	----	22.2	---
PI	6.2	85.4%	2.6	88.3%
Adaptive H_∞	8.1	80.0%	4.8	78.4%

Table 4 Jitter Control Subsystem Performance Comparison for Multiple Varying Frequencies

The power spectral density plots below show the frequency spectrum of the error signal for each controller. Figure 38 shows the power spectral density for the first set of frequencies, 12 Hz and 38 Hz, and Figure 39 shows the power spectral density of the second set of frequencies, 9 Hz and 46 Hz.

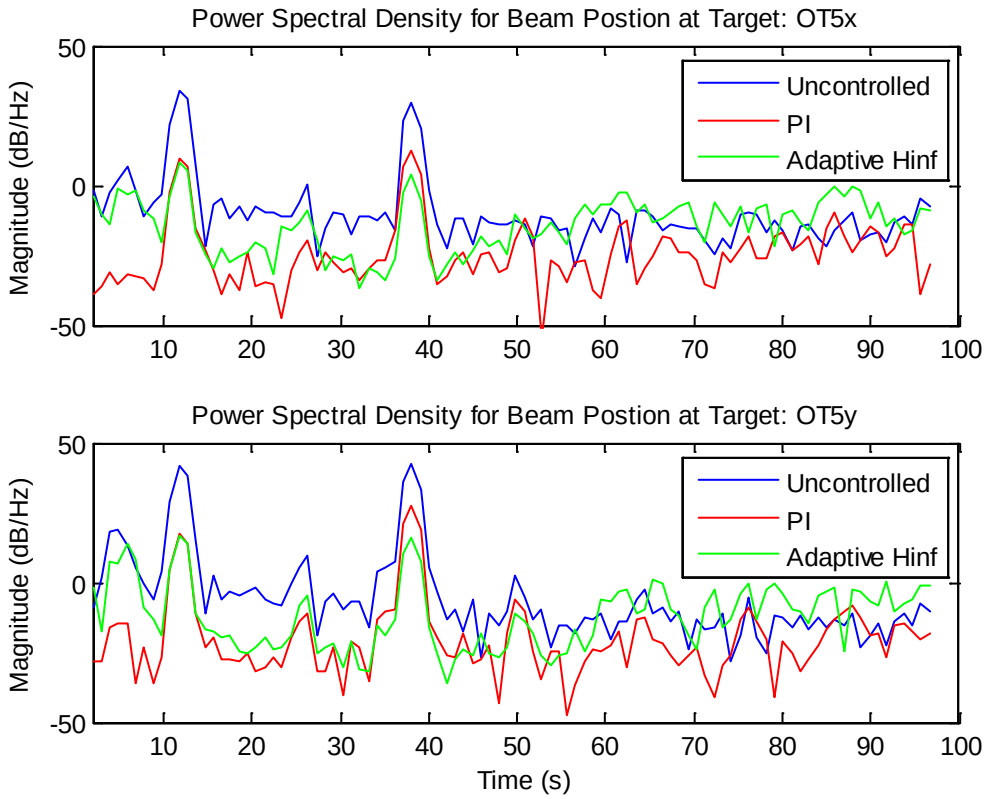


Figure 38 Power Spectral Density Comparison for 12 Hz and 38 Hz

When reviewing Figure 38 it is apparent that for the two disturbance frequencies the adaptive H_∞ controller actually outperforms the PI controller. It is important to note that the

resonant 5 Hz tone is apparent in the system and while the PI controller mitigated that signal the H_∞ was set to N = 2 frequencies, so it only identified the two largest disturbances. The performance was similar at the 12 Hz signal with both controllers obtaining about 26 dB/Hz reduction, but at the 38 Hz tone the adaptive H_∞ controller achieved roughly 10 dB/Hz more attenuation in each axis than the PI controller.

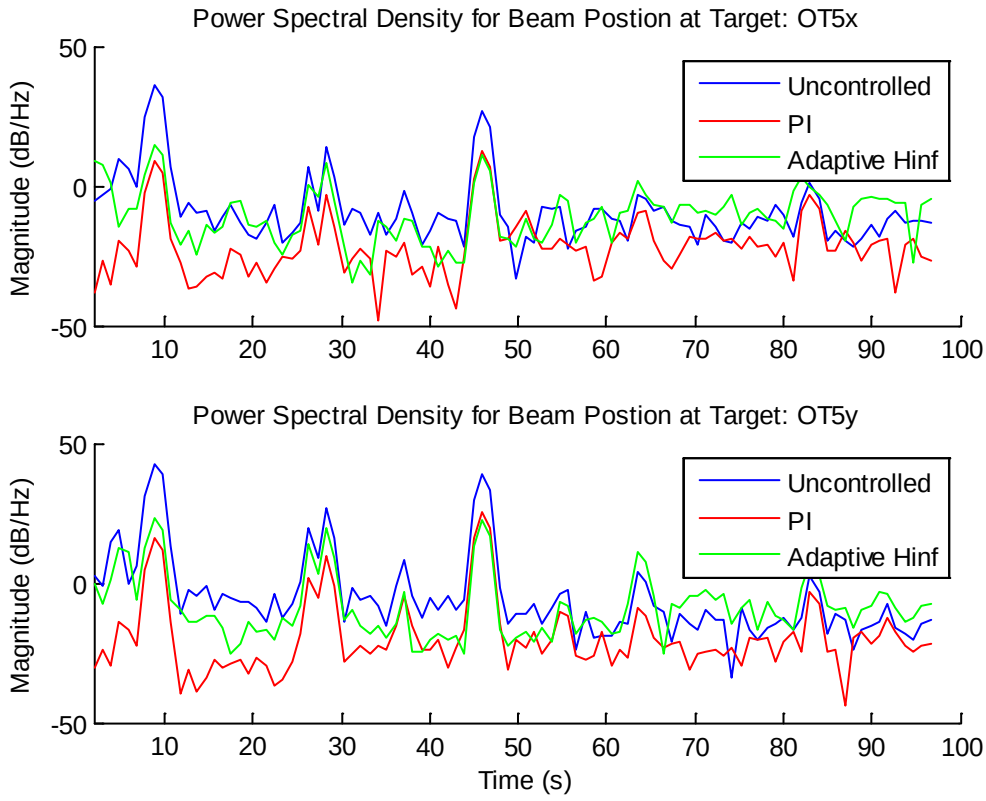


Figure 39 Power Spectral Density Comparison for 9 Hz and 46 Hz

Figure 39 shows the power spectral density comparison for the second set of frequencies, 9 Hz and 46 Hz. The controllers both show significant signs of reduction in the tonal frequencies, but the PI performs better, achieving approximately 6.5 dB/Hz more attenuation at the 9 Hz signal, while they both achieve about 16 dB/Hz attenuation at the 46 Hz disturbance. The 5 Hz tone is present again, and as stated before the PI attenuates it, but the H_∞ does not as it was set to identify and attenuate 2 frequencies. The main cause for the adaptive H_∞ controller achieving less attenuation, when compared to the first set of frequencies may be due to how the controller calculates its desired amount of attenuation. When the frequencies shift, the old controller is still

in effect for 1.024 seconds while it identifys the new frequencies. During that period, since the frequencies were within a few Hz, there was some spillover attenuation occurring, so it decreased the desired amount of attenuation for the new controller. When the new controller is calculated and implemented, the amount of reduction is thrown off because the effects of the previous controller are now gone.

4.2 PI Target Tracking Controller

The target tracking controller implements a series of two PI controllers for each axis. The first PI controller allows the system to follow the velocity initially, but due to the ramp up there is a slight lag. The second PI controller helps to remove that lag, while the first moves the beam at the same velocity as the target.

The target tracking controller is tested with the same target motion that will be used to test the whole beam control system, but without the jitter element. The target tracking controller is tested for stability and to identify any adverse effects it may have on the jitter control portion of the beam control system. Figure 40 shows the local beam position on the moving target.

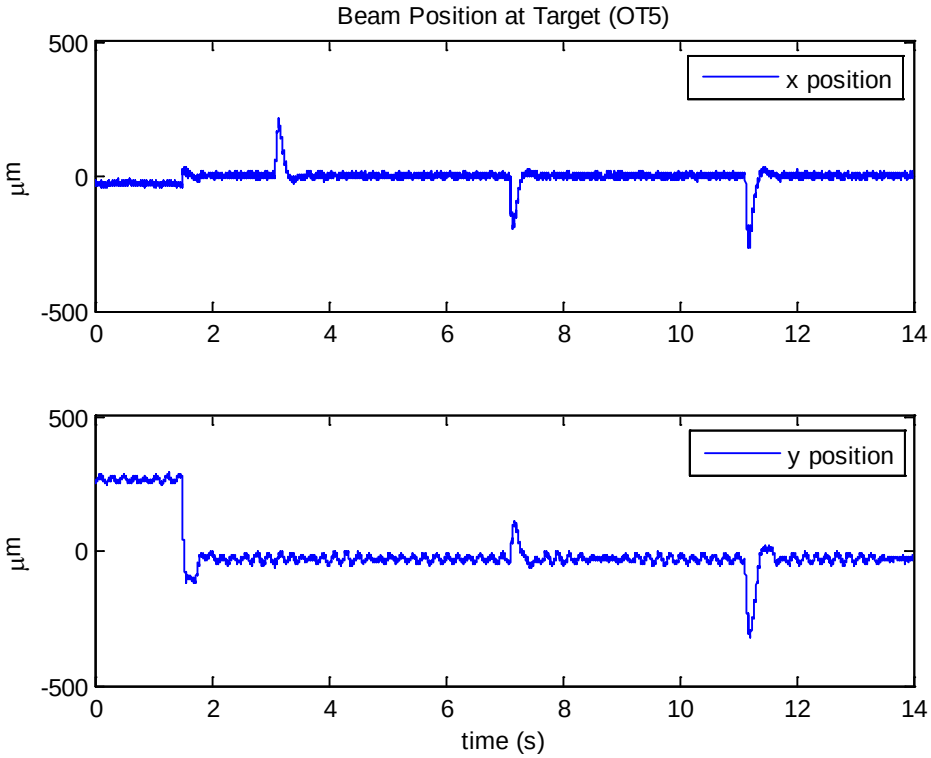


Figure 40 Beam Position with PI Target Tracking Controller

The beam position plot shows the distinct points where the target begins moving and when the velocity changes. The spikes that appear in the beam position occur when the component velocity in the respective axis changes. At 3.15 seconds the target begins moving in the negative x-direction and since only the x-axis has changed velocity the spike only occurs on the x position. At 7.15 seconds the target stops in the x-direction and begins moving in the negative y-directions, then at 11.15 seconds the target moves in both the positive x and y-direction. The movements at 7.15 seconds and 11.15 seconds cause spikes in both axes as the velocity components in both directions are effected at the respective time marks. After each velocity change the controller takes 0.2 seconds to settle and reacquire the target center.

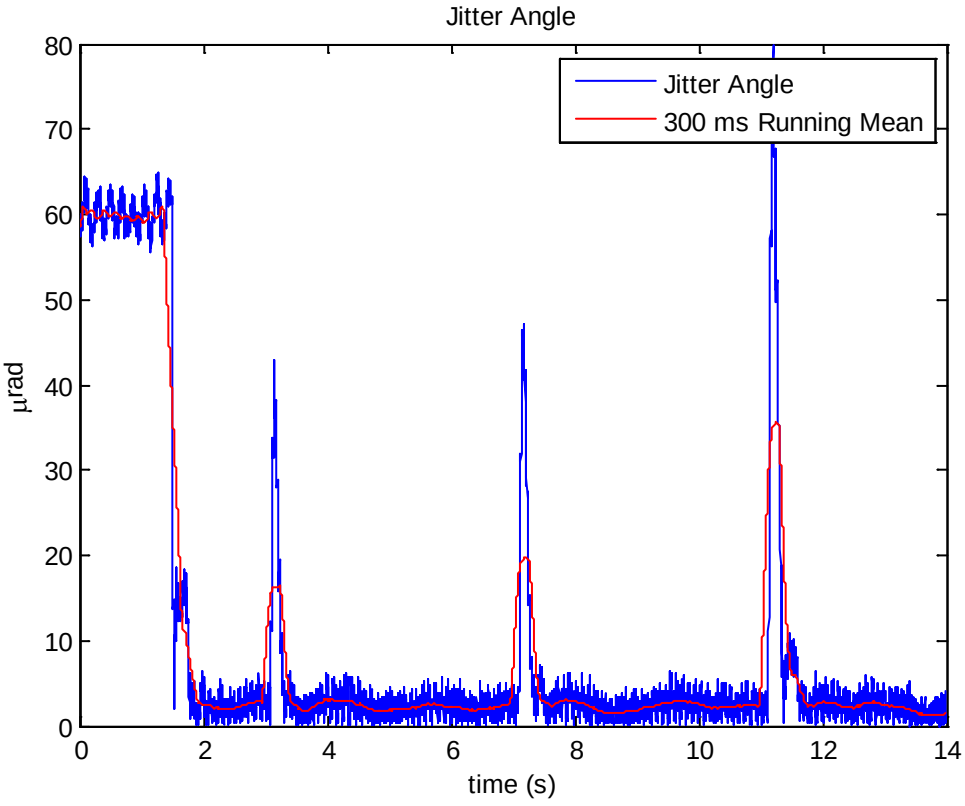


Figure 41 Jitter Angle and 300 ms Running Mean of PI Target Tracking Controller

Figure 41 shows the jitter angle of the target tracking controller, and similar to the beam position plots in Figure 40, there are spikes at the times when the velocity changes. The second two spikes, at 7.15 seconds and 11.15 seconds, are larger than the first spike due to velocity vector changing in both axes, unlike the first where only the x-direction changes. The RMS of

the jitter angle for the entire controlled run is 9.2 µrad. The standard deviation of the jitter angle is 8.3 µrad. The target tracking controller was not optimally designed to handle acceleration, any change in velocity, and a better tuned or more advanced controller can eliminate those spikes. If the spikes due to the velocity changes are removed, then the RMS jitter angle reduces to 2.6 µrad and the standard deviation of the jitter angle reduces to 1.1 µrad

The focus of this research was to provide an adaptive controller that could actively identify and attenuate jitter while engaging a moving target. Tracking was a secondary objective of this research. To properly analyze the effectiveness of the jitter controller the sections unaffected by the large peaks from the tracking controller need to be analyzed, along with the entire run.

4.3 Total Beam Control System Performance

4.3.1 Beam Control System With PI Jitter Control

With both jitter controllers tested and evaluated the next step is to compare the entire system. The target tracking controller is the same for both jitter controllers, therefore there is no need to analyze it separately and it has already been tested and tuned for stability. The gain values for the target tracking system remain constant in both tests, as well as the original jitter controllers. The same disturbance frequencies and magnitudes are entered into the system, at the same time periods. The target motion consists of three components, testing x and y-axis, both independently and combined.

Figure 42 shows the relative beam position at the target with the target motion and induced jitter. Similar to the jitter only test, the PI jitter controller appears to have a significant effect on the beam’s position, and mitigates most of the jitter

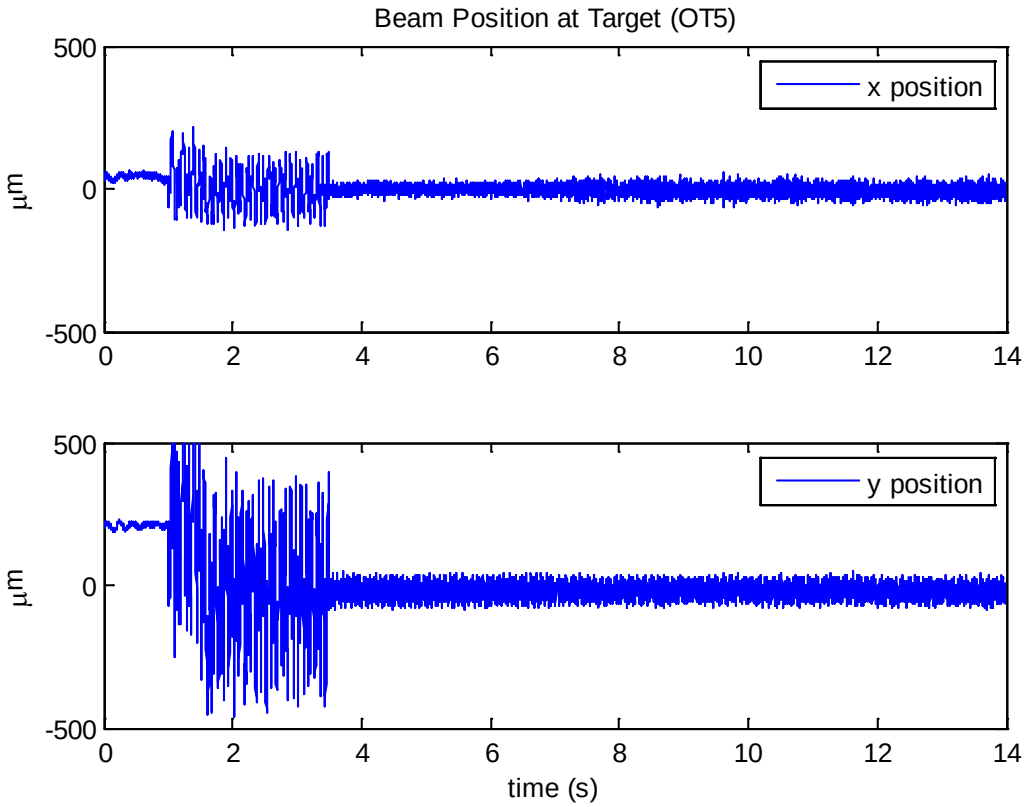


Figure 42 Beam Position with PI Jitter Control and Target

The disturbance starts at the 1 second mark. At 1.5 seconds after run start the target tracking controller turns on, and centers the beam on the stationary target. At 3.5 seconds after run start the jitter controller turns on, removing a majority of the jitter, and the target begins to move following the motion defined in the experimental procedures section.

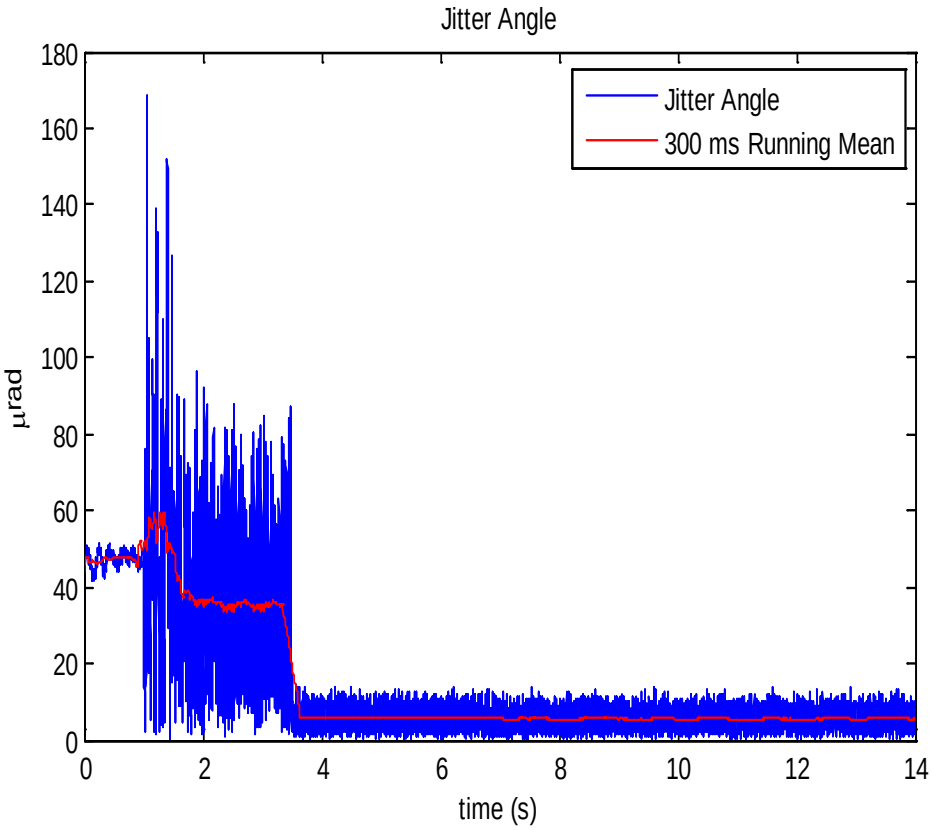


Figure 43 Jitter Angle and 300 ms Running Mean for PI Jitter Controller with Moving Target

Figure 43 show the jitter angle and 300 ms running mean for the PI jitter controller with varying frequencies and target tracking. The RMS of the jitter angle is reduced 85.4% to 6.2 µrad, achieving nearly the same reduction as the jitter test alone. The standard deviation of the jitter angle is reduced 88.3% to 2.6 µrad.

4.3.2 Adaptive H_∞ Controller with Target Tracking

The adaptive H_∞ controller was then tested with the same disturbances, target motion, and timing. Figure 44 shows the beam position of the laser on the target. The controller provides significant improvement in the effects of the jitter, while the target tracking controller accurately follows the motion of the target.

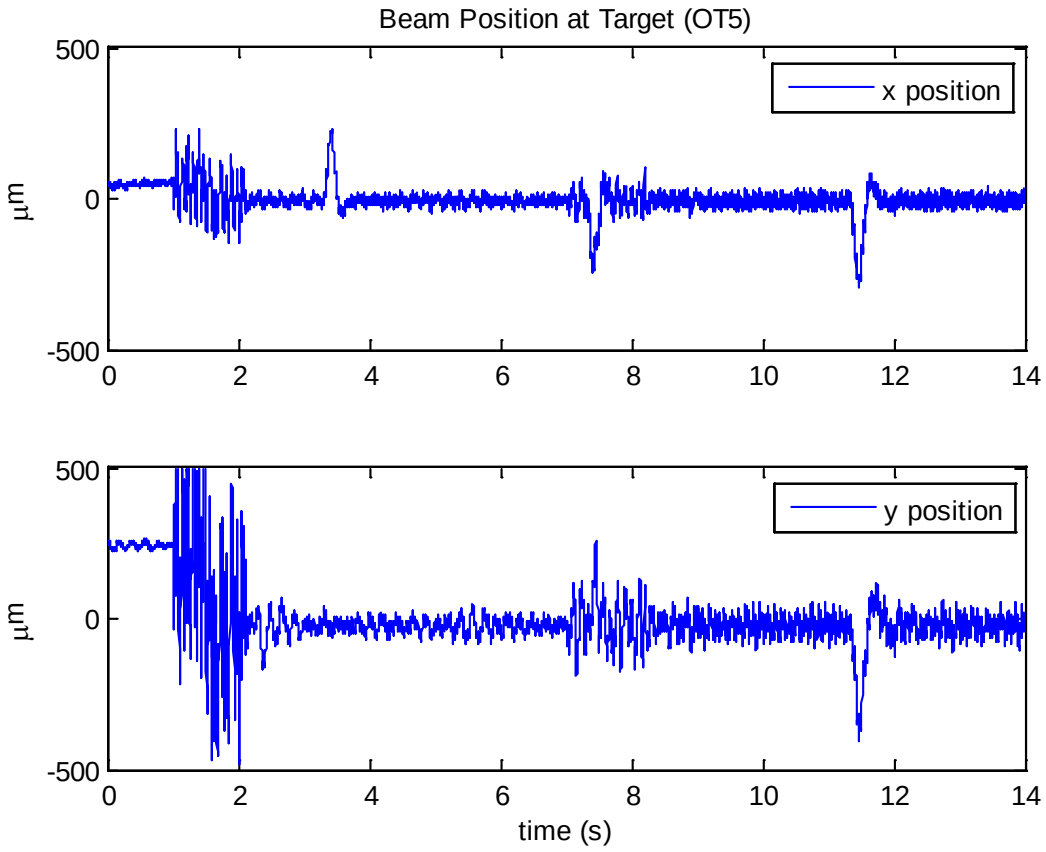


Figure 44 Beam Position with Adaptive H∞ Control and Target Tracking

The target tracking begins at 1.5 seconds after run start, as seen when it centers the beam on the target initially. Then at 2.024 seconds the H_∞ initiates and begins attenuating the jitter. The target begins to move at 3.5 seconds after run start, as seen in the spike of the x-position. The various spikes in the position plot are due to the change in velocity in a given axis, and the target tracking controller having to adjust accordingly. Also at the 7 second after run start mark the change in frequencies is visible while the controller is identifying them.

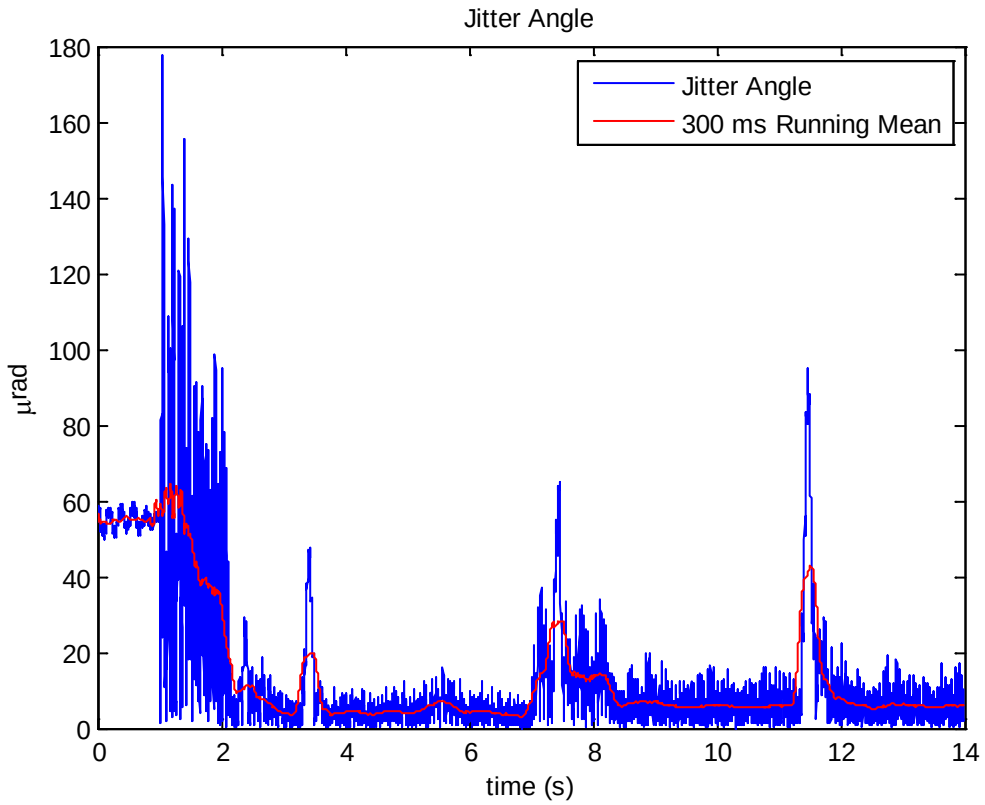


Figure 45 Jitter Angle and 300 ms Running Mean for H∞ Controller with Target Tracking

Figure 45 shows the jitter angle and 300 ms running mean for the adaptive H_∞ controller with a moving target. The spikes from the position plot carry over significantly due to the magnitude. During the periods of constant velocity the adaptive H_∞ controller provides great control. The RMS of the jitter angle for the entire length of control is reduced 70% to 12.9 µrad. The standard deviation of the jitter angle for the entire length of control is reduced 54.5% to 10.1 µrad. If the spikes due to velocity changes and the tracking controller are removed the RMS of the jitter angle is reduced 81.5% to 7.8 µrad and the standard deviation is reduced 78.8% to 4.7 µrad.

The removal of disturbance caused by the target tracking controller is a realistic assumption given that our target tracking controller was not designed to adequately handle the changes in velocity. A PI controller with added tuning or another specific controller meant to handle the sharp changes in velocity exist and can be implemented in a way to avoid the magnitude of the spikes experienced during this research in the beam position. With that

assumption the Adaptive H_∞ controller proved to be effective at removing the jitter with a moving target.

4.3.3 Beam Control System Comparison

Table 5 shows the performance characteristics of each control system and the amount of improvement with respect to the uncontrolled jitter runs. The PI jitter controller and adaptive H_∞ controller reduce the jitter, but the PI outperforms the H_∞ controller with respect to the whole run. When the continuous velocity portions are analyzed for the H_∞ controller, thereby removing the errors of the target tracking controller, the amount of reduction is similar to that of the PI jitter controller.

Control Scheme	**RMS Jitter Angle (μrad)**	**Improvement in RMS Jitter Angle**	**Standard Deviation (μrad)**	**Improvement in Standard Deviation**
Uncontrolled	42.2	----	22.2	---
PI	6.2	85.4%	2.6	88.3%
Adaptive H_∞	12.9	70.0%	10.1	54.5%
Adaptive H_∞ (Constant Velocity)	7.8	81.5%	4.7	78.8%

Table 5 Comparison of Beam Control Systems for Jitter Control With Target Tracking

Table 6 shows the RMS jitter angle and standard deviation of the jitter angle for both cases with and without target tracking. This allows an analysis into whether the target tracking controller had any negative impact on the jitter controller. The prediction was the target tracking would have little to no effect on the jitter control due to the high and low-pass filters that differentiate the lower frequency motion and higher frequency jitter. The only significant effect was during the transition between velocities with the H_∞ controller, but during the times of constant velocity there were negligible effects.

Control Scheme	RMS Jitter Angle w/o Target Motion (µrad)	RMS Jitter Angle w/ Target Motion (µrad)	Standard Deviation w/o Target Motion (µrad)	Standard Deviation with Target Motion (µrad)
PI	6.2	6.2	2.6	2.6
Adaptive H_∞	8.1	12.9	4.8	10.1
Adaptive H_∞ (Constant Velocity)	8.1*	7.8	8.1*	4.7

Table 6 Beam Control System Comparison With and Without Target Motion

***Used same values for regular H_∞ controller

Figures 46 and 47 show the power spectral density of the beam control systems for the uncontrolled jitter run, as well as the two runs with target tracking using adaptive H_∞ and PI jitter controllers.

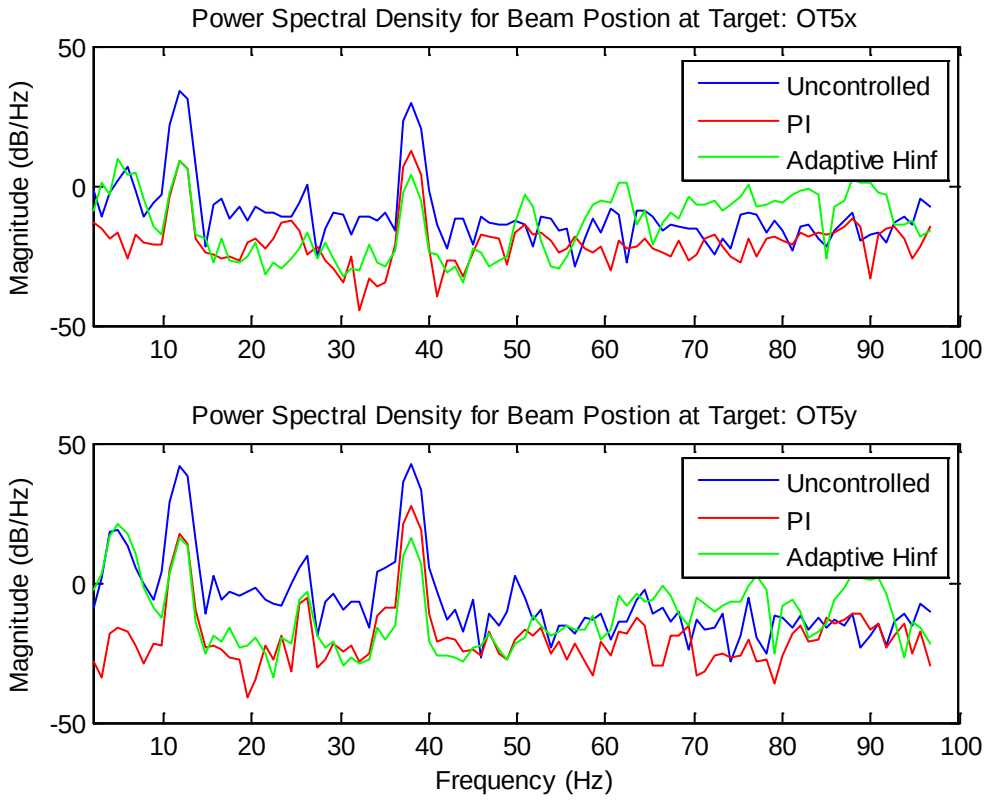


Figure 46 Power Spectral Density of Beam Control Systems with Target Motion and First Set of Frequencies [12 Hz and 38 Hz]

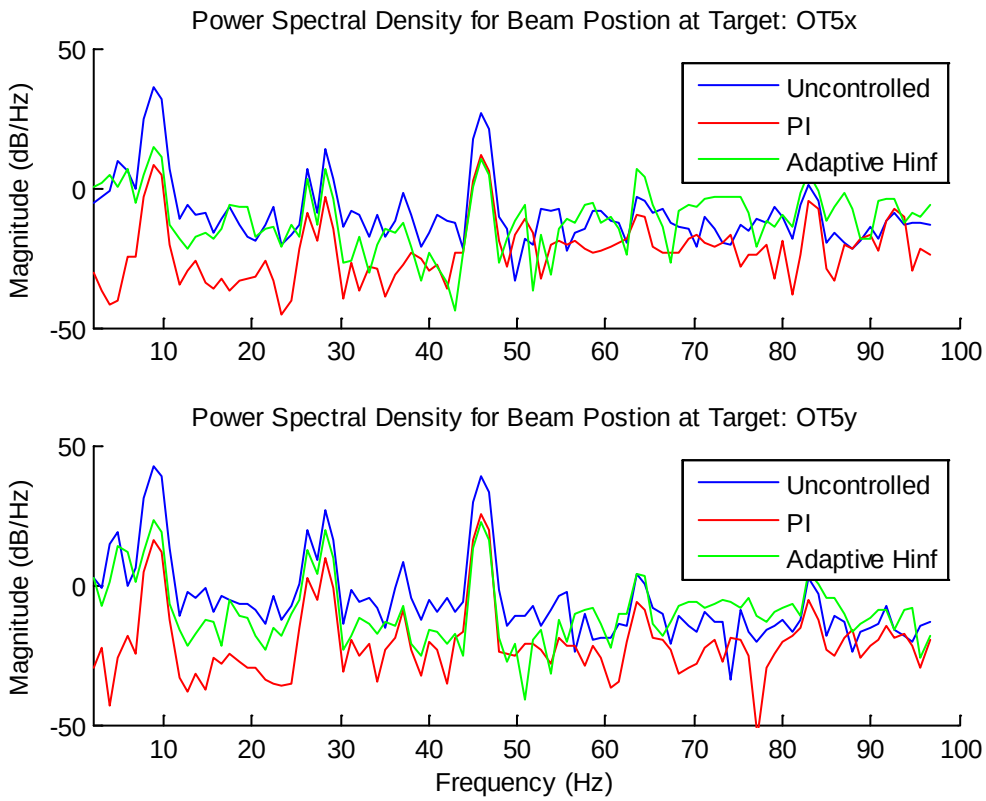


Figure 47 Power Spectral Density of Beam Control Systems with Target Motion and Second Set of Frequencies [9 Hz and 46 Hz]

The results for the power spectral density plots reinforce the data in Table 6, in that the target motion did not significantly affect the ability of the jitter controller to work properly and identify the frequencies. Similar to the jitter control tests without target motion, in the first set of frequencies seen in Figure 44 the adaptive H_∞ controller has higher attenuation at the disturbance frequencies. Then, in the second set of frequencies, the PI jitter controller performs slightly better. The problem identified with the H_∞ controller identifying new frequencies and the amount of desired attenuation being distorted due to the slight attenuation caused by the old controller in the jitter only test is present in in this experiment as well.

Although the H_∞ is obtaining better attenuation at the specific frequencies, the PI jitter controller still performs better overall with the jitter angle measurements. This is due to the fact the PI controller also deals with the 5 Hz resonant frequency of the platform, which the H_∞ ignores because it is set to identify only two frequencies. The second cause for the better overall

performance of the PI controller is that the H_∞ causes some slight spillover increase at the higher frequencies as an effect of the bode squeeze theorem.

4.3.4 Beam Control System Error Analysis

There are several significant errors that exist in this research. The mechanical structure of the beam control system is the largest source of error. Even though research grade materials are used, there are tolerances built in that are similar to the level of precision we are trying to achieve; µm and µrad. These tolerances, along with the desired level of accuracy for the system, lead to an inability to accurately align the components perfectly. Misalignment causes small movements in one axis as the other is commanded to move, thus when the individual controllers attempt to control a disturbance in one axis, they add a small disturbance to the opposing axis. A compounding source of error for this spillover of movements into the opposing axis is not accounting for the local platform roll in the controller. When the platform rolls, the reference axes of the controller shifts with respect to the target, so now a movement in what the controller thinks is the absolute x-axis will be seen as movement in both the x and y-axis at the target.

Another source of error in the system is the resolution at which the sensors and actuators operate. They have resolutions and repeatability values on the µrad level and since that is the desired accuracy of the system, that can induce a significant amount of error into the system. Along with the resolutions of the system, the signals and commands from the sensors and actuators are converted from voltages to distances and back again when they arrive or leave system. These conversion factors are calculated numbers from numerical test, for example applying a voltage to a FSM, reading the movement, and creating scaling factor for voltage to µrad. Any form of conversion and measurements from testing will induce small error into the system which accumulate as more conversions are added throughout the beam control system

Additionally there are many assumptions that are made about the system, such as ideal FSM movement with no mechanical cross-coupling and a plant model with instantaneous response. Some of the assumptions are made in order to make controller calculations possible, and while H_∞ is designed for uncertainties, they still add some error into the system while it remains stable. There is also an issue with the commanded position of the mirror and the actual

resulting position since it is an open-loop system with no feedback control on the actual mirror command itself.

5. Conclusion

5.1 Results

This research successfully created an adaptive H_∞ control algorithm that in real-time, could identify frequencies of disturbance and create a stabilizing controller, all while engaging a moving target. The PI jitter controller, as a baseline comparison to classical control theory, provided consistently greater than 80% reduction in both RMS and standard deviation of the jitter angle for multiple, shifting frequencies with and without target motion. The adaptive H_∞ control algorithm provided greater than 78% reduction in RMS and standard deviation of the jitter angle with no target motion. With the target motion the performance was evaluated to two parts: the entire controlled run and when the system is evaluated excluding the error cause by the changing velocities which the target tracking controller was not designed to optimally handle. When the entire length of the run is evaluated the adaptive H_∞ controller achieves a 54% reduction in the standard deviation of the jitter angle and a 70% reduction in the RMS of the jitter angle. When the effects of the changing velocities are removed, the adaptive H_∞ controller achieved greater than 78% reduction in both RMS and standard deviation of the jitter angle. The fact the target tracking controller does not experience any noticeable overshoot when the PI jitter controller is used, it could possibly mean that either PI controller needs to be added to the weight for the adaptive H_∞ controller or possibly a third order integrator term needs to be added to the PI target tracking controller. The results also confirm that jitter control and target motion can be treated as two different ranges of frequencies, thus allow it to be controlled with two separate loops. In both jitter controllers, when there was target motion, the difference in amount of reduction seen in the jitter angle versus no target motion run was negligible. The adaptive H_∞ controller was set to only identify two frequencies, so the tonal 5 Hz platform frequency was left to pass which induced some error which the PI jitter controller had controlled

The adaptive H_∞ proved it was capable of identifying both multiple and shifting frequencies, in real-time, and had the ability to successfully attenuate those frequencies. The adaptive H_∞ was not significantly affected in its ability to mitigate jitter when engaging a target

at a constant velocity. The adaptive H_∞ is a practical means for jitter control in a directed energy weapon and is capable of improving its effectiveness.

5.2 Future Work

Though the adaptive control algorithm did prove effective through this research, there are significant improvements that can be made to obtain results that outperform the PI jitter controller. The original concept was to perform a real-time optimization of the dampening ratios for the weights so the desired reduction at every frequency was achieved from the final weight so larger magnitude values could be used. Without optimization narrower weights were required, so there was minimal spillover gain at outside frequencies, but in order to do so much small values had to be used in the weights. Using these smaller values add poles to the system which are much closer to the imaginary axis, which can lead to marginal stability. In simulation the Nelder-Mead minimization method performed very well, but the issue was the efficiency of the algorithm and when it was implemented on the real-time environment it required slightly too much calculation time. Creation of a more efficient optimization method is a primary area of future work.

Another area for future work is the ability to determine in real-time the reference frames of the source platform and target, with respect to each other, so that a transformation matrix can be created. This transformation matrix can be used to adjust the commands for the FSMs so that there are no cross-coupling effects while steering the beam as a result of rolling the platform.

Although feedback for the position of the beam on the target is an ideal, from a controls perspective, it is not always available on each platform in real-world environments. Applications like the ABL have the ability to accurately determine the beam position at tactical distances, but the next step is to implement this controller with a beam prediction based error feedback system.

An improved target tracking controller is also an important aspect for future research.

APPENDIX A: Newport Fast Steering Mirrors

FSM System

	FSM-300	FSM-320
Number of Axes	2 (tip-tilt)	2 (tip-tilt)
Angular Range from ±10 V	± 26.2 mrad (± 1.5°), mechanical(1)	± 26.2 mrad (± 1.5°), mechanical(1)
Resolution	≤ 1 µrad rms, mechanical(1)	≤ 1 µrad rms, mechanical(1)
Repeatability	≤ 3 µrad rms, mechanical(1)	≤ 3 µrad rms, mechanical(1)
Accuracy From ±26.2 mrad, 20°C(1,2)	≤ 0.262 mrad (0.015°), mechanical(1)	≤ 0.262 mrad (0.015°), mechanical(1)
Linearity From ±26.2 mrad, 20°C(1,2)	≤ 1.0%	≤ 1.0%
Closed-Loop Amplitude Bandwidth(2) (-3 dB)	≥ 800 Hz at 10 mV	≥ 350 Hz at 10 mV
Closed-Loop Phase Bandwidth(2) (60° lag)	≥ 400 Hz	≥ 325 Hz
Response Flatness(2)	Peaking ≤ 3 dB	Peaking ≤ 3 dB
Noise Equivalent Angle (1 Hz to 10 kHz)	≤ 3 µrad rms	≤ 3 µrad rms
Resolution of Local Position Sensor	≤ 0.5 µrad	≤ 0.5 µrad
Quiescent Power at FSM Assembly	≤ 5 W at any angle ± 26.2 mrad	≤ 5 W at any angle ± 26.2 mrad
Operating Temperature Range(2)	0 to 35°C (32 to 95°F)	0 to 35°C (32 to 95°F)
Storage Temperature Range	-20 to 55°C (-4 to 131°F)	-20 to 55°C (-4 to 131°F)
Warm-up Time for Mirror Stability(2) at 20°C	≤ 10 minutes	≤ 10 minutes
Mirror Thermal Drift(2)	≤ 5 µrad/°C, mechanical(1)	≤ 5 µrad/°C, mechanical(1)
Optical Axis Location	1.5 in. (38.1 mm) high, centered left-to-right	1.5 in. (38.1 mm) high, centered left-to-right
Mirror Head Weight with Base	15.3 oz (434 g)	15.3 oz (434 g)
Interconnect Cable Length	9.8 ft (3 m)	9.8 ft (3 m)

Standard Mirror Options

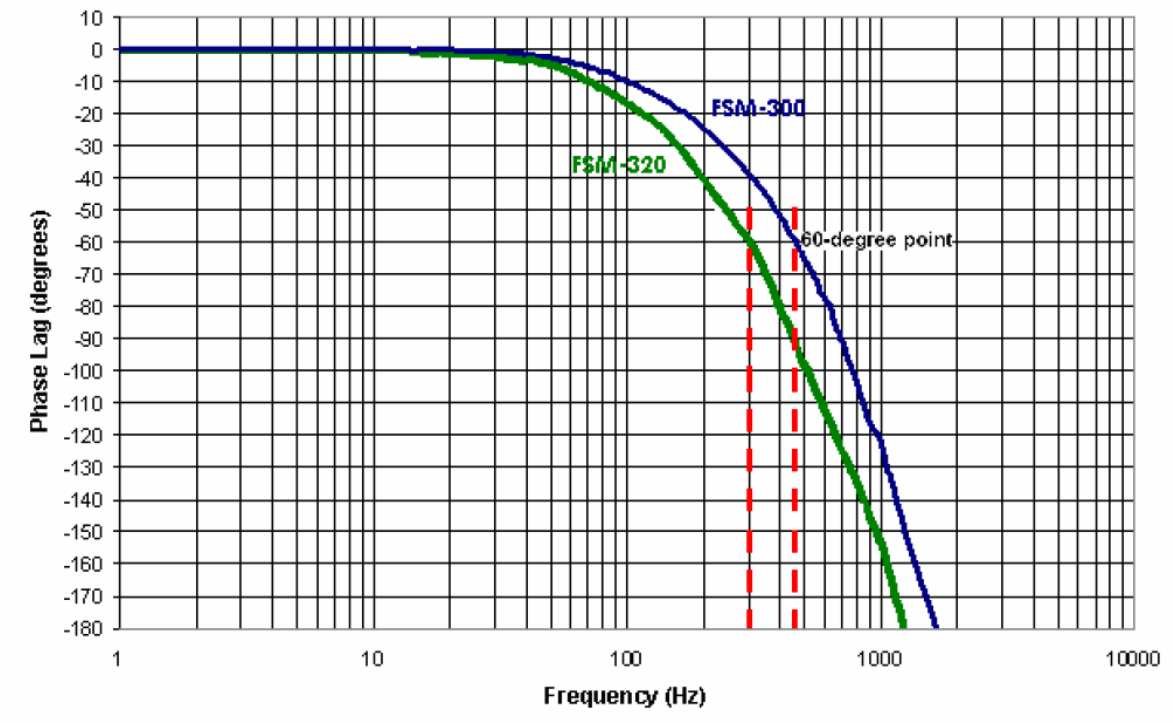
	FSM-300	FSM-320
Mirror Substrate Material	Pyrex	Fused Silica
Mirror Retaining Mechanism	Mirror bonded to aluminum carrier (user replaceable).	Mirror bonded to stainless steel carrier (replaceable).
Mirror Pivot Point (centered on mirror)	Gimbaled 12.19 mm behind mirror surface	Gimbaled 9.15 mm behind mirror surface
Mirror Diameter	25.4 mm	50.8 mm
Mirror Thickness	6.0 mm	3.0 mm
Mirror Wedge	≤ 5 arc min	≤ 5 arc min
Clear Aperture(3) at 0° angle of incidence	≥ 20.3 mm	≥ 40.6 mm
Clear Aperture(3) at 45° angle of incidence	≥ 14.4 mm	≥ 28.8 mm
Surface Flatness(3) (after coating and bonding)	≤ λ/10 at 632.8 nm over clear aperture	≤ λ/2 at 632.8 nm over clear aperture
Surface Quality(3)	15-5 scratch-dig	40-20 scratch-dig
Reflectivity, Standard Coatings(3)		
ER.1 Coating: Enhanced Aluminum	> 93%, 450-700 nm	> 93%, 450-700 nm
ER.4 Coating: Protected Gold	> 96%, 650- 1700 nm; > 98% from 1.7-2.0 µm	Please contact Newport.
Additional coating options	Please contact Newport.	Please contact Newport.

FOOTNOTES:

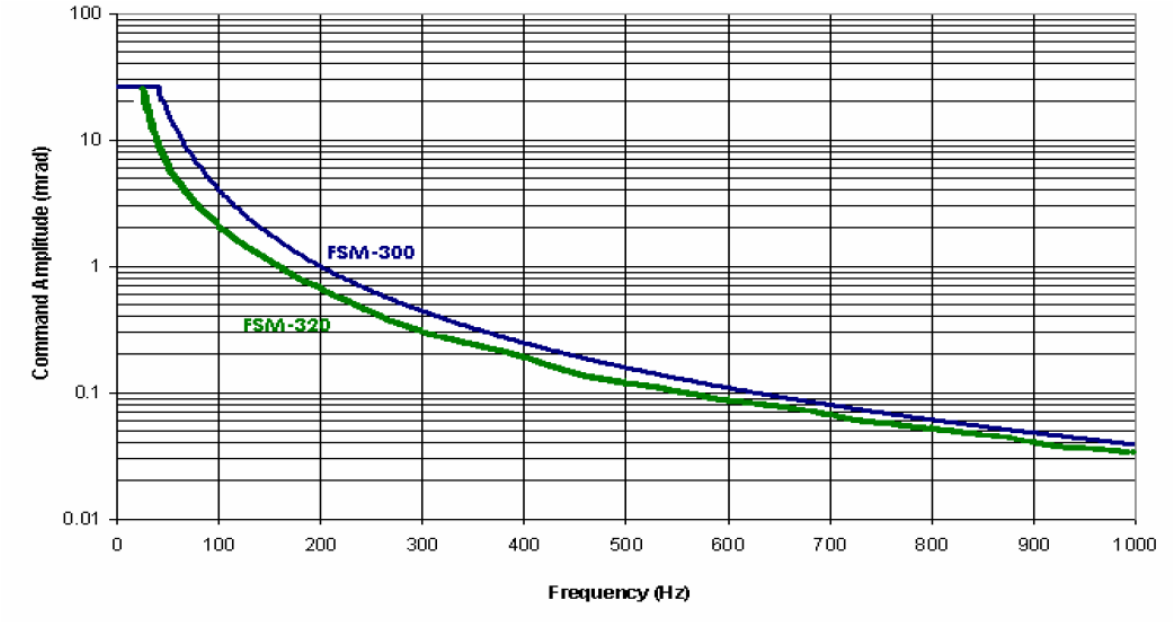
1) Optical angular range is equal to twice the mechanical angular range.
2) Measured under position output control. Optical closed-loop performance is also determined by external feedback electronics.
3) Optical parameters apply to central 80% of mirror aperture.

FSM-CD300B Controller/Driver

Command Input and Position Output	Analog, ±10 V = ±26.2 mrad
Peak Operating Power to Mirror	30 W
Continuous Max Operating Power to Mirror	15 W
Thermal Protection	60°C at mirror coil
Operating Temperature[2]	0 to 35°C (32 to 95°F)
Storage Temperature	-20 to 55°C (-4 to 131°F)
Use Location	Indoor use only
Relative humidity	< 95%, non-condensing
Operating altitude	< 3,000 m (10,000 ft)
Power	100-240 Vac ±10%, 47-63 Hz
Current consumption (typical)	0.40 A @ 100 Vac, 0.25 A @ 240 Vac
Fuses	2 ea, “slo-blo” (T), 5 x 20 mm, rated 2.5 A, 250 Vac
Weight	5.5 lbs (2.5 kg)
Case Dimensions (excluding connectors)	3.9” x 9.0” x 10.0” [h x w x d] (100 x 229 x 254 mm)



Typical phase angle Bode plot for small-angle angle excitation.
Amplitude 0.262 mrad.



Typical shut-down curve as a function of amplitude and frequency at 20°C. Continuous operation is "safe" below the line. Derate for higher ambient temperatures.

APPENDIX B: Aerotech Inc. Linear Motor Actuator

LMA/LMAC Series SPECIFICATIONS

LMA										
Total Travel	100 mm	200 mm	300 mm	400 mm	500 mm	600 mm	700 mm	800 mm	900 mm	1000 mm
Drive System	Linear Brushless Servomotor									
Feedback	Noncontact Linear Encoder									
Resolution[5]	5 nm - 1.0 µm									
Max Travel Speed[1][6]	5 m/s									
Max Linear Acceleration[6]	3 g (30 m/s^2 no load)									
Max Horizontal Load[2]	40 kg									
Max Side Load[2]	20 kg									
Continuous Force[3]	207 N									
Continuous Force (20 psi)[3]	276 N									
Peak Force[3]	1106 N									
Accuracy	±1 µm/25 mm									
Calibrated Accuracy[4]	±5 µm Over Entire Length of Travel									
Repeatability	±0.5 µm									
Nominal Stage Weight	16.3 kg	18.5 kg	20.7 kg	22.8 kg	25.0 kg	27.2 kg	29.4 kg	31.6 kg	33.7 kg	35.9 kg
Moving Weight	3.28 kg (142 Motor Option)					5.18 kg (264 Motor Option)				
Material	Aluminum									
Finish	Electroless Nickel Base/Black Hardcoat Tabletop									

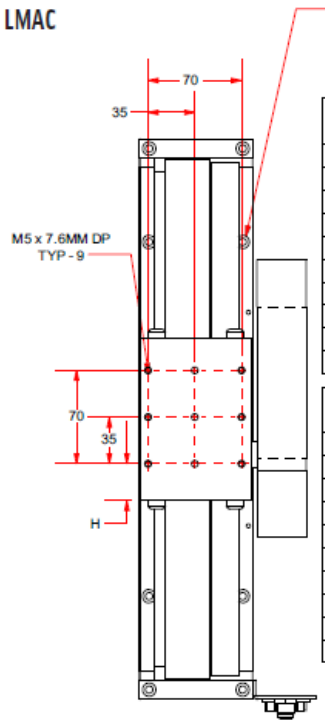
Notes:
1. Maximum speed based on stage capability; maximum application velocity may be limited by system data rate and resolution.
2. Maximum load based on bearing capability; maximum application load may be limited by acceleration requirements.
3. Specifications based on BLM-264-A motor. Travel increases by 20 mm when using the BLM-142-A motor.
4. Available with Aerotech controller.
5. May require encoder multiplier.
6. Consult factory on high speed and/or high acceleration applications.

LMAC										
Total Travel	100 mm	200 mm	300 mm	400 mm	500 mm	600 mm	700 mm	800 mm	900 mm	1000 mm
Drive System	Linear Brushless Servomotor									
Feedback	Noncontact Linear Encoder									
Resolution[5]	5 nm - 1.0 µm									
Max Travel Speed[1][6]	5 m/s									
Max Linear Acceleration[6]	3 g (30 m/s^2 no load)									
Max Horizontal Load[2]	20 kg									
Max Side Load[2]	10 kg									
Continuous Force[3]	47.9 N									
Continuous Force (20 psi)[3]	73.1 N									
Peak Force[3]	292 N									
Accuracy	±1 µm/25 mm									
Calibrated Accuracy[4]	±5 µm Over Entire Length of Travel									
Repeatability	±0.5 µm									
Nominal Stage Weight	4.5 kg	5.4 kg	6.3 kg	7.2 kg	8.1 kg	9.0 kg	9.9 kg	10.8 kg	11.7 kg	12.6 kg
Moving Weight	0.79 kg (95 Motor Option)					1.00 kg (143 Motor Option)				
Material	Aluminum									
Finish	Electroless Nickel Base/Black Hardcoat Tabletop									

Notes:
1. Maximum speed based on stage capability; maximum application velocity may be limited by system data rate and resolution.
2. Maximum load based on bearing capability; maximum application load may be limited by acceleration requirements.
3. Specifications based on BLMUC-143-A motor. Travel increases by 35 mm when using the 95 motor.
4. Available with Aerotech controller.
5. May require encoder multiplier.
6. Consult factory on high speed and/or high acceleration applications.

LMAC Series DIMENSIONS

LMAC

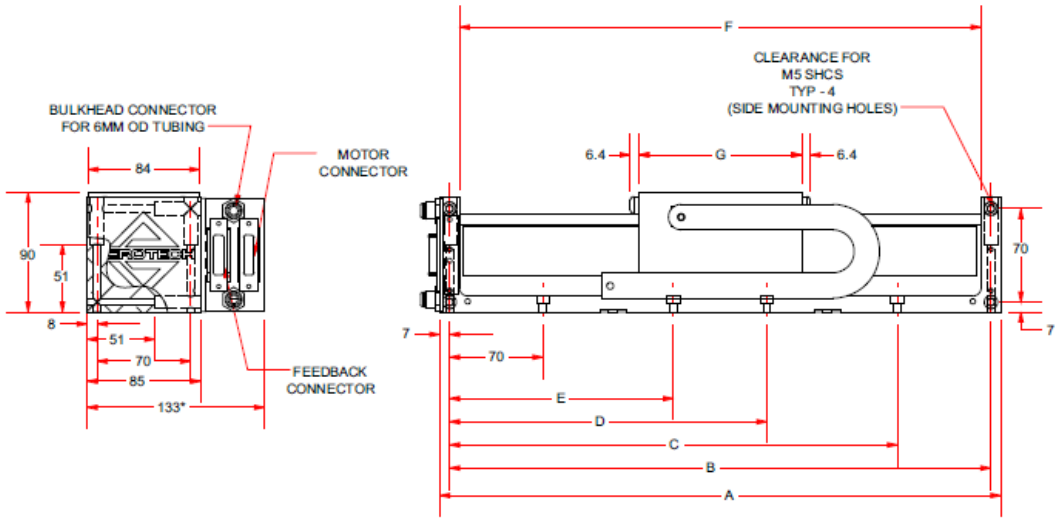


BASIC MODEL	TOTAL TRAVEL	DIMENSIONS - MILLIMETERS							
		A	B	C	D	E	F	G	H
LMAC100	100	320	305	235	188	118	291	162	46
LMAC200	200	420	405	335	238	168	391	162	46
LMAC300	300	520	505	435	288	218	491	162	46
LMAC400	400	620	605	535	338	268	591	162	46
LMAC500	500	720	705	635	388	318	691	162	46
LMAC600	600	820	805	735	438	368	791	162	46
LMAC700	700	920	905	835	488	418	891	162	46
LMAC800	800	1020	1005	935	538	468	991	162	46
LMAC900	900	1120	1105	1035	588	518	1091	162	46
LMAC1000	1000	1220	1205	1135	638	568	1191	162	46

BASIC MODEL	TOTAL TRAVEL	DIMENSIONS - MILLIMETERS							
		A	B	C	D	E	F	G	H
LMAC135	135	320	305	235	188	118	291	122	28
LMAC235	235	420	405	335	238	168	391	122	28
LMAC335	335	520	505	435	288	218	491	122	28
LMAC435	435	620	605	535	338	268	591	122	28
LMAC535	535	720	705	635	388	318	691	122	28
LMAC635	635	820	805	735	438	368	791	122	28
LMAC735	735	920	905	835	488	418	891	122	28
LMAC835	835	1020	1005	935	538	468	991	122	28
LMAC935	935	1120	1105	1035	588	518	1091	122	28
LMAC1035	1035	1220	1205	1135	638	568	1191	122	28

* ADD 40 TO WIDTH FOR B3/B4 CONNECTOR OPTIONS

RIGHT HAND MODEL SHOWN



DIMENSIONS: MILLIMETERS

APPENDIX C: CSA Engineering Inertial Actuator

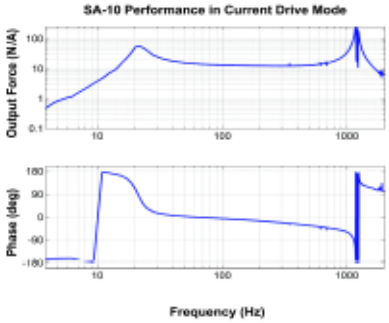
SA Series Inertial Actuators

CSA Engineering, Inc.

- Inertial force generator
- 1-10 lbf broadband output (0 - peak)
- Peak outputs greater than 100 lbf
- Wide bandwidth (20 to 1000 Hz)
- Self contained

SA series actuators deliver inertial force over a wide bandwidth in compact, rugged, electromagnetically efficient forms. The actuators use an electromagnetic circuit with a moving magnet that allows the coil to be thermally grounded to the housing. Magnets are suspended by specially designed long life flexures. The force is generated along the axis of the cylindical units.

Typical applications include active damping or vibration cancellation, mounts for active vibration isolation, or disturbance generation. Dynamic amplification at frequencies near the actuator resonance results in large force outputs. A rigid housing enables direct insertion of the SA into structural load paths.



Actuators are specified by force capacity and internal suspension resonance. Standard options/accessories include alternative end caps, coils of specified impedance, a variety of cable interfaces, and current or voltage drive mode amplifiers. The overall design is easily customizable to meet the requirements of mounting configurations, drive electronics, or mass budgets.

The SA1, SA5, and SA10 are standard products. Also available are the SA2, SA35 and other non-standard models. Actuators are specified as SA*x*-*f* where *x* is the zero-peak force output at high frequency in pounds and *f* is the primary resonant frequency in Hz. For example, the SA5-60 produces 5 pounds force and has a 60 Hz resonance.

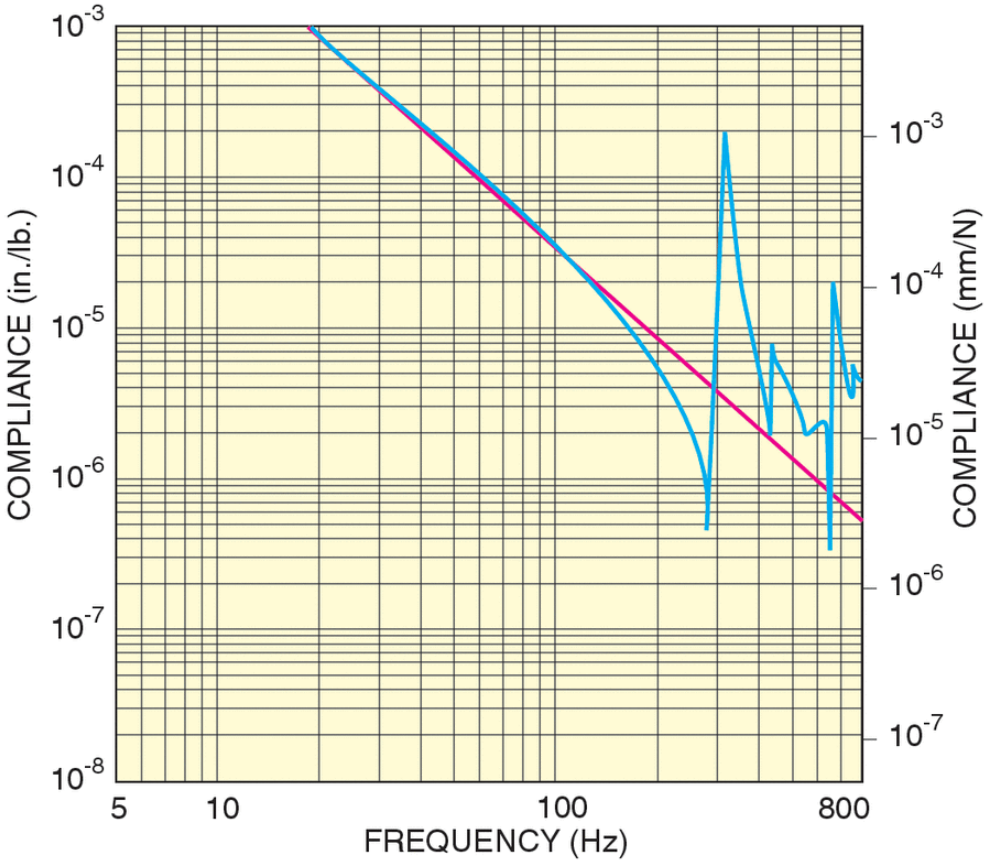
SPECIFICATIONS				
	SA1	SA5	SA10	Units
Rated Force Output*	1	5	10	lbf (0-peak)
Bandwidth	40-1000	20-1000	20-1000	Hz
Motor Constant**	0.5	2	5	lbf/Amp
Resonant Frequency***	30-200	30-200	30-200	Hz
Resistance**	2	2	2	Ohm
Total Mass	0.25	2.9	5.5	lbm
Diameter	35	76	93	mm
Height	30	66	92	mm

* Significantly greater forces possible with good heatsinking
** Typical values
*** User-specified. Manufactured to ± 2-3%

For more information, email *actuators@csaengineering.com*

APPENDIX D: Newport Breadboard

Model	**IG-33-2**
Width	3 ft.
Length	3 ft.
Thickness	2.3 in. (58 mm)
Thread Type	1/4-20
Mounting Hole Pattern	1.0 in. grid
Surface Flatness	±0.006 in. over 2 ft. (±0.15 mm over 600 mm)
Working Surface	400 Series ferromagnetic stainless steel
Core Design	Trussed honeycomb, vertically bonded closed cell construction, 0.010 in. (0.25 mm) Steel sheet materials, 0.030 in. (0.76 mm)
Broadband Damping	Integrated Damping including constrained layer core, damped working surface and composite edge finish
Mounting Hole Type	Cut (not rolled) threads with countersink
Hole/Core Sealing	Easy clean conical cup 0.75 in. (19 mm) deep Non-corrosive high impact polymer material
Maximum Dynamic Deflection Coefficient	$<17 \times 10^{-4}$
Maximum Relative Motion Value	$<13 \times 10^{-7}$ in. ($<3.3 \times 10^{-5}$ mm)
Deflection Under Load	$<15 \times 10^{-5}$ in.
Top and Bottom Skins	0.134 (3.4 mm) thick with integrated damping layer


COMPLIANCE (in./lb.)
10-3
10-4
10-5
10-6
10-7
10-8
5
10
100
800
FREQUENCY (Hz)
COMPLIANCE (mm/N)
10-3
10-4
10-5
10-6
10-7

APPENDIX E: On-Trak PSD

PSD Theory (http://www.on-trak.com/theory.html)

Description

Position Sensing Detectors “PSD’s” are silicon photodiodes that provide an analog output directly proportional to the position of a light spot on the detector active area. The PSD allows you to simultaneously monitor position and light intensity. The PSD is a continuous analog position sensor. Compared to discrete element devices, the PSD offers outstanding position linearity, high analog resolution, fast response time, and simple operating circuits.

Theory Of Operation

A Position Sensing Detector consists of n-type silicon substrate with two resistive layers separated by a p-n junction. The front side has an ion implanted p-type resistive layer with two contacts at opposite ends. The back side has an ion implanted n-type resistive layer with two contacts at opposite ends placed orthogonally to the contacts on the front side. On a single axis PSD, the electrodes are placed at opposite ends of the p-type resistive layer. A light spot within the spectral range of silicon will generate a photocurrent that flows from the incident point through the resistive layers to the electrodes. The resistivity of the ion implanted layer is extremely uniform so the photogenerated current at each electrode is inversely proportional to the distance between the incident spot of light and electrodes. The PSD outputs track the motion of the “centroid of power density” to an extremely high resolution and ultra-high linearity. On-Trak Position Sensing Amplifiers take the photocurrent from each electrode and process the signals to provide X, Y outputs independent of light intensity.

Position Resolution

The position resolution of a PSD is the minimum detectable displacement of a spot of light on the detector surface. The position resolution of On-Trak PSDs are proven better than one part in a million. Resolution dependent on:

- Detector Size
- Detector Noise
- Light Input Intensity
- Bandwidth of the Electronic Signal Processing Circuits

Position Linearity

Position non-linearity is defined as geometric position error divided by detector length and is measured within 80% of the detector length. Position non-linearity is typically better than 0.05% for the single axis PSD and better than

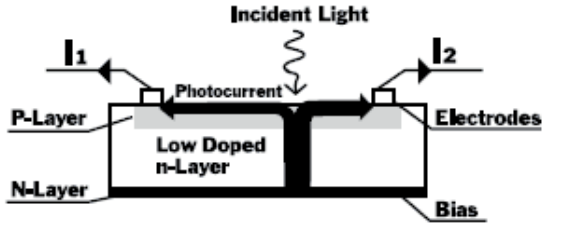
0.3% for the duolateral. The On-Trak vs competitor two-dimensional linearity plot shows the ultra linear characteristic of these PSDs.

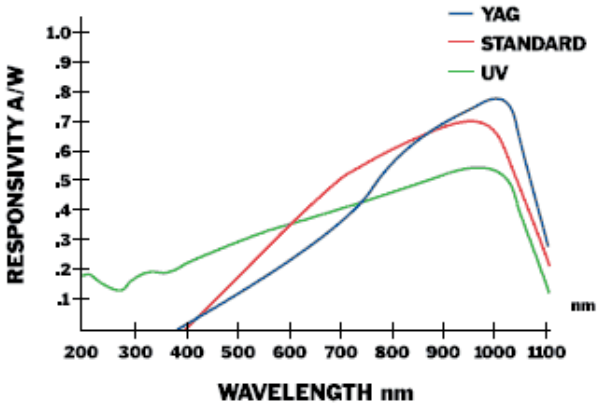
One-Dimensional PSD

The one-dimensional PSD detects a light spot moving over its surface in a single direction. The photoelectric current generated by the incident light flows through the device and is seen as an input bias current divided into two output currents. The distribution of the output currents show the light position on the detector.

Duolateral Two-Dimensional PSD

The duolateral two-dimensional PSD detects an incident light spot position on its square surface. The photoelectric current generated by the incident light flows through the device and is seen as two input currents and two output currents. The distribution of the output currents show the light position of one dimension (Y), and the distribution of the input currents show the light position of the second dimension (X).

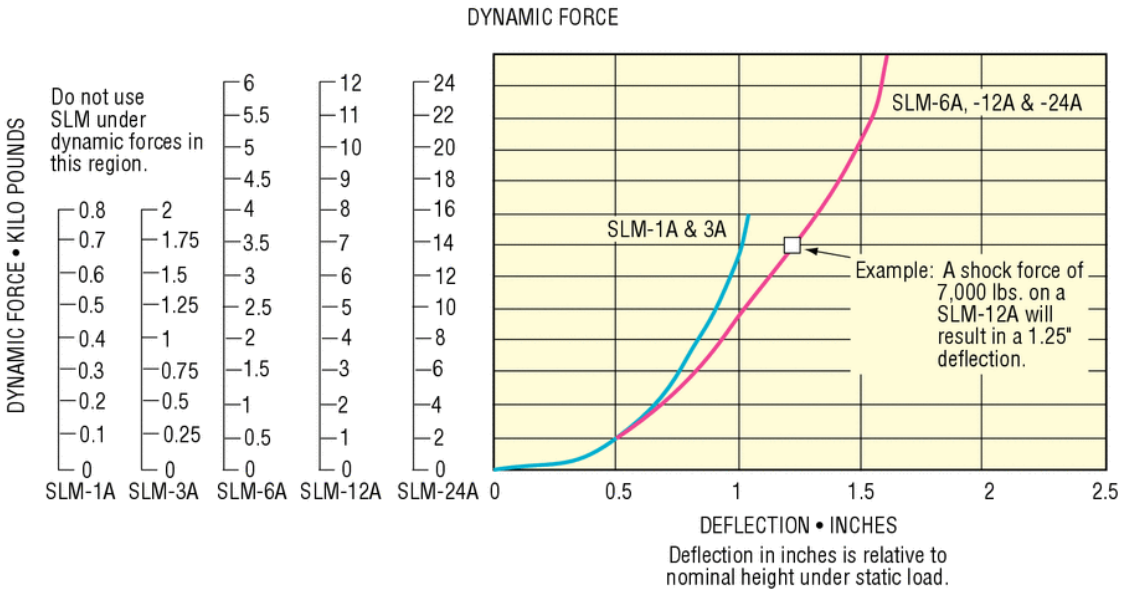


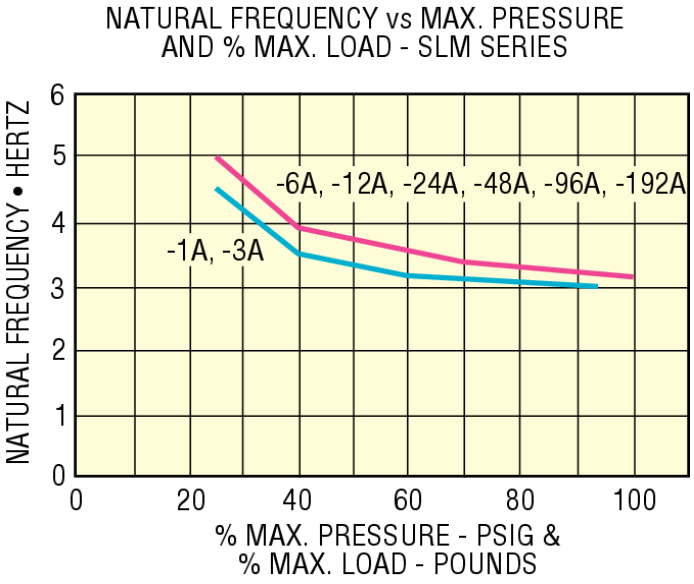


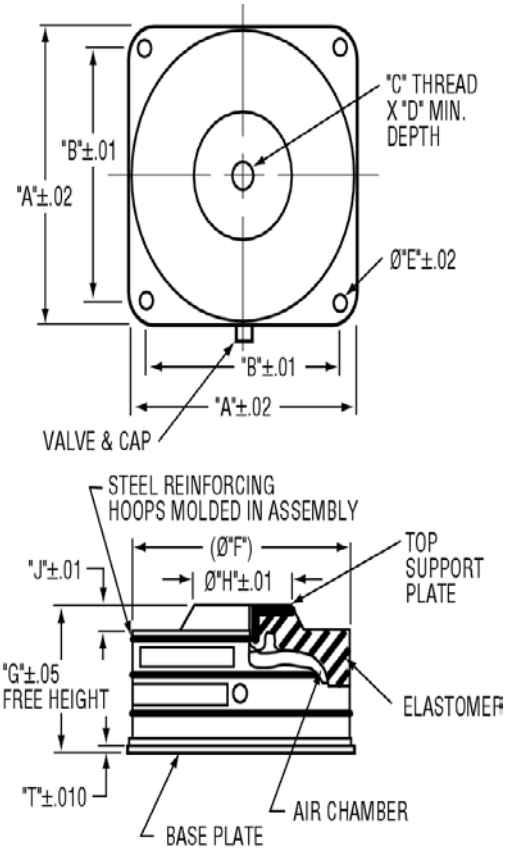
PSD Type	Spectral Range	Responsivity
Standard	400-1100 nm	0.70 A/W @ 940 nm

APPENDIX F: Newport Compact Air-Mount

Model	SLM-3A
Load per Isolator	136 kg (300 lb)
Load Capacity	136 kg (300 lb)
Max. Air Pressure	60 psi
Natural Frequency (Nominal), Max.	5
Natural Frequency (Nominal), Min.	3 Hz
Isolator Weight	0.68 kg (1.5 lb)
Operating Temperature Range	-40 to 83 °C

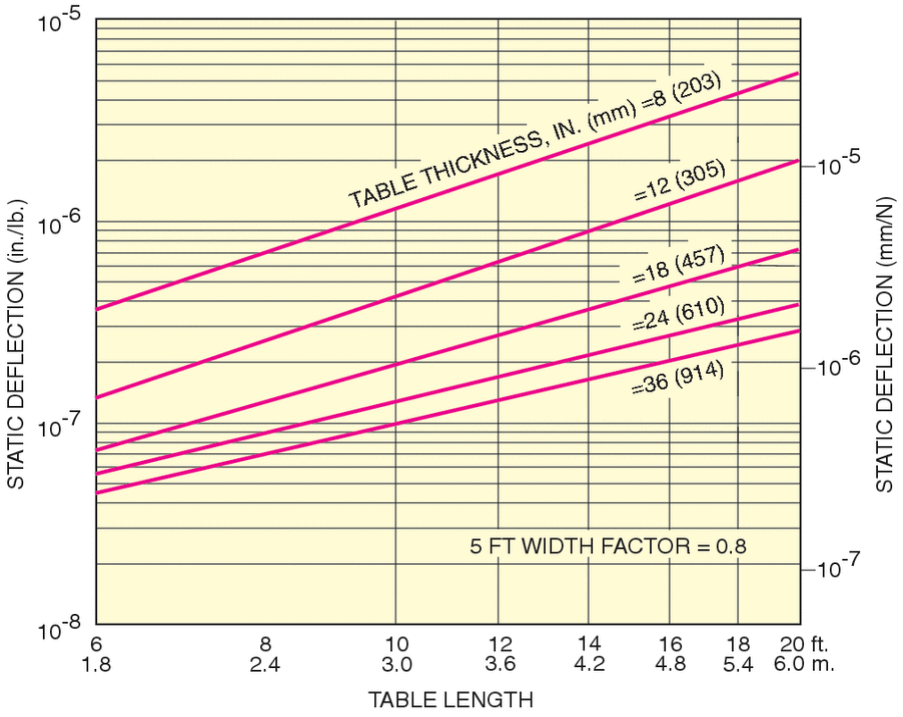


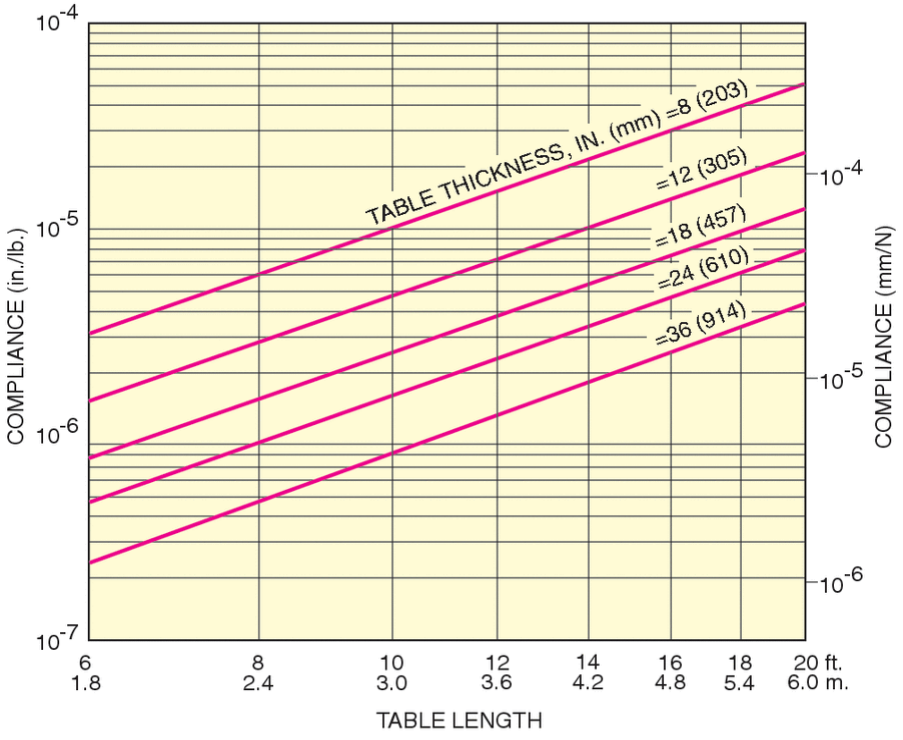

NATURAL FREQUENCY vs MAX. PRESSURE
AND % MAX. LOAD - SLM SERIES
NATURAL FREQUENCY • HERTZ
0
1
2
3
4
5
6
-6A, -12A, -24A, -48A, -96A, -192A
-1A, -3A
0
20
40
60
80
100
% MAX. PRESSURE - PSIG &
% MAX. LOAD - POUNDS

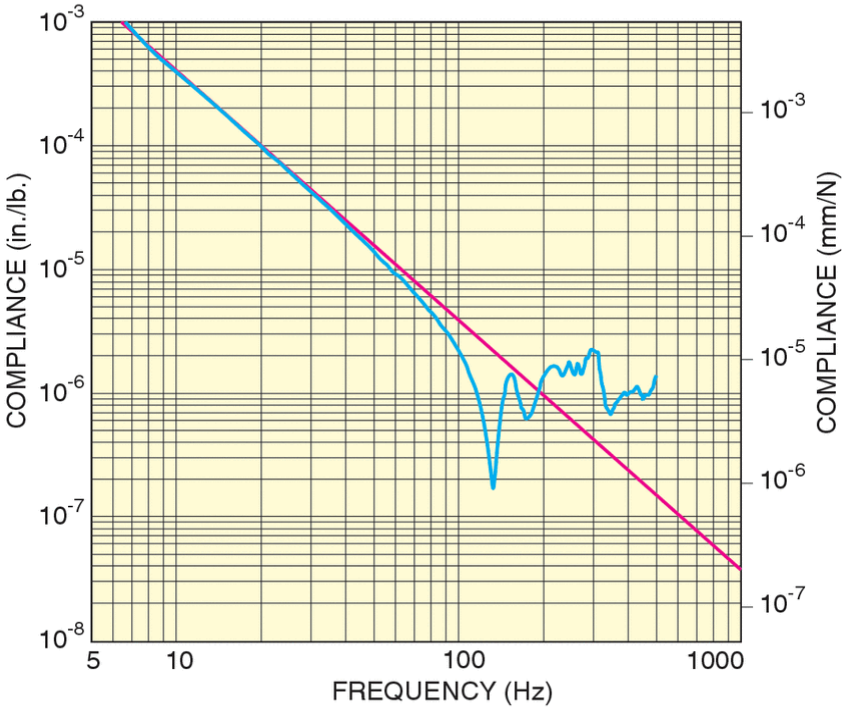

"C" THREAD
X "D" MIN.
DEPTH
"B"±.01
"A"±.02
Ø"E"±.02
"B"±.01
"A"±.02
VALVE & CAP
STEEL REINFORCING
HOOPS MOLDED IN ASSEMBLY
(Ø"F")
Ø"H"±.01
TOP
SUPPORT
PLATE
"J"±.01
"G"±.05
FREE HEIGHT
ELASTOMER
"T"±.010
AIR CHAMBER
BASE PLATE

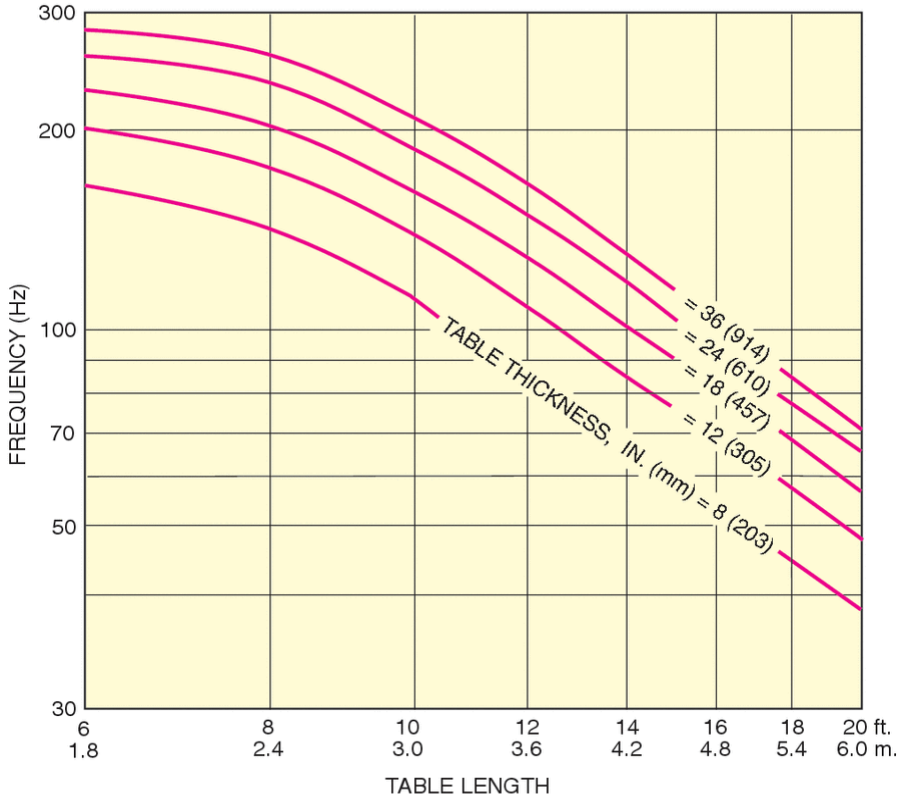
APPENDIX G: Newport Optical Tables

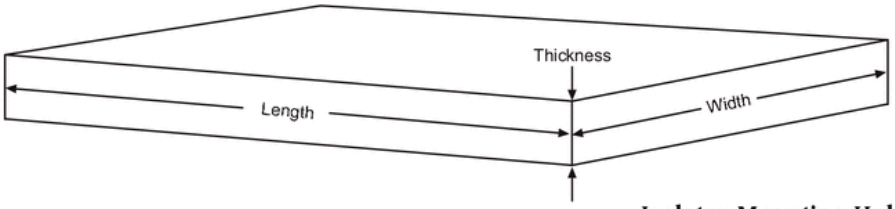
Model	RS4000-48-8
Mounting Hole Type	1/4-20
Mounting Hole Pattern	1 in. grid
Length	8 ft.
Width	4 ft.
Thickness	8 in.
Working Surface	400 series ferromagnetic stainless steel
Deflection Under Load	$<5.0 \times 10^{-5}$ in. in. ($<1.3 \times 10^{-3}$ mm mm)
Maximum Dynamic Deflection Coefficient	0.4×10^{-3}
Core Design	Trussed Honeycomb, Vertically Bonded Closed Cell Construction, 0.010 in. Steel sheet materials, 0.030 in. triple core interface
Broadband Damping	Constrained layer core, damped working surface and composite edge finish
Hole/Core Sealing	Easy clean conical cup 0.75 in. (19 mm) deep, Non-corrosive high impact polymer material
Top and Bottom Skins	3/16 in. (4.8 mm) thick with integrated damping layer
Crated Weight	521 kg (1172 lb)

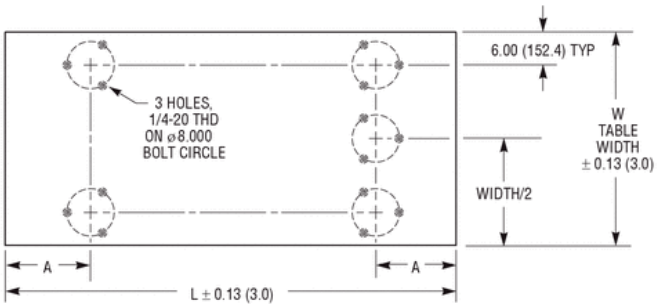

STATIC DEFLECTION (in./lb.)
STATIC DEFLECTION (mm/N)
TABLE THICKNESS, IN. (mm)
=8 (203)
=12 (305)
=18 (457)
=24 (610)
=36 (914)
5 FT WIDTH FACTOR = 0.8
TABLE LENGTH
6 1.8
8 2.4
10 3.0
12 3.6
14 4.2
16 4.8
18 5.4
20 ft. 6.0 m.


COMPLIANCE (in./lb.)
COMPLIANCE (mm/N)
TABLE THICKNESS, IN. (mm)
=8 (203)
=12 (305)
=18 (457)
=24 (610)
=36 (914)
TABLE LENGTH
6 1.8
8 2.4
10 3.0
12 3.6
14 4.2
16 4.8
18 5.4
20 ft. 6.0 m.


COMPLIANCE (in./lb.)
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
COMPLIANCE (mm/N)
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
5
10
100
1000
FREQUENCY (Hz)


FREQUENCY (Hz)
300
200
100
70
50
30
TABLE THICKNESS, IN. (mm)
= 36 (914)
= 24 (610)
= 18 (457)
= 12 (305)
= 8 (203)
6
1.8
8
2.4
10
3.0
12
3.6
14
4.2
16
4.8
18
5.4
20 ft.
6.0 m.
TABLE LENGTH

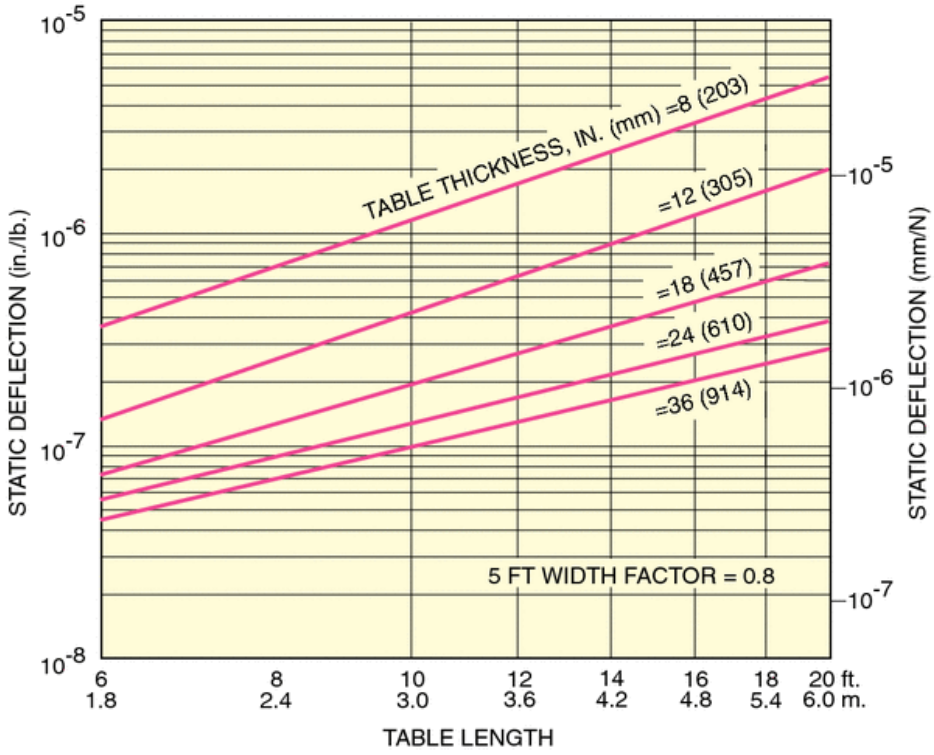


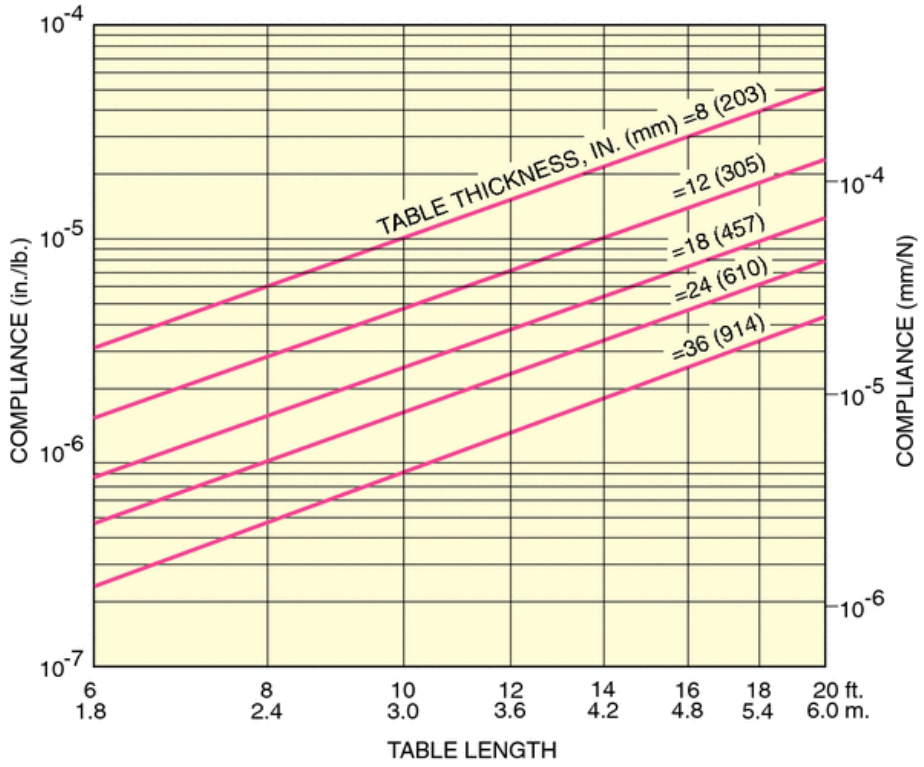


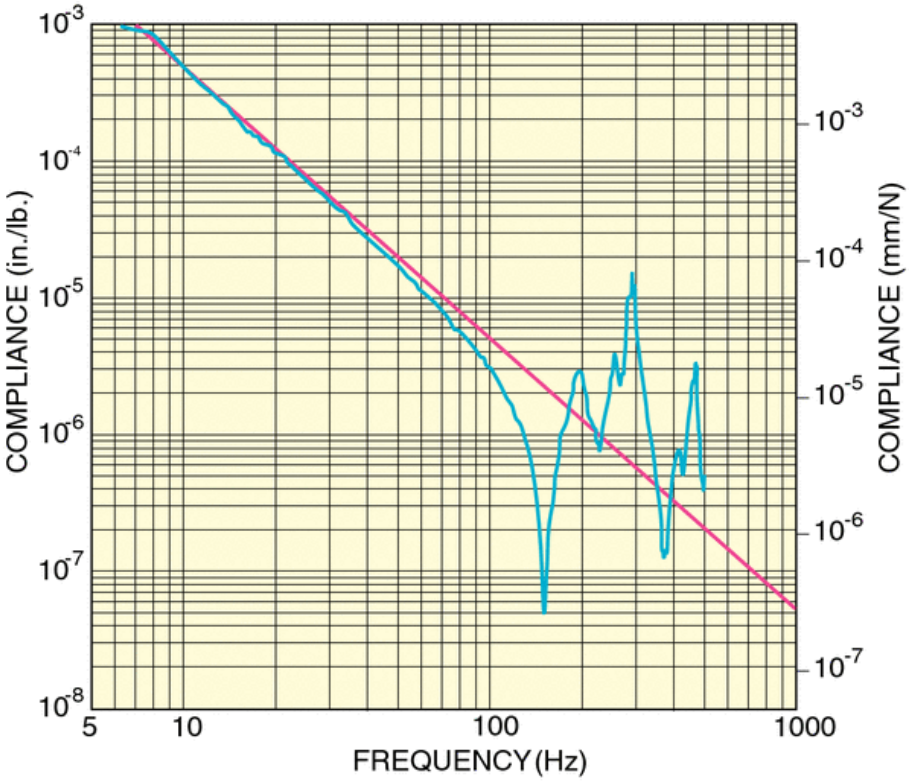
Isolator Mounting Hole Dimensions

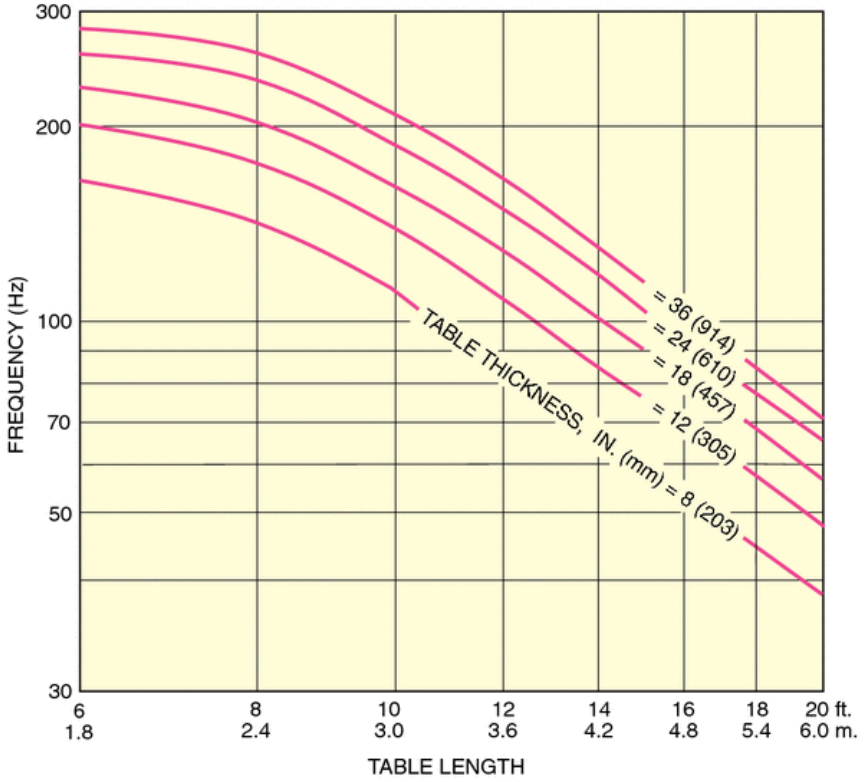
WIDTH (W)	LENGTH (L)	DIMENSION (A)
in.	ft	in.
36.0	20.0	53.0
48.0	18.0	47.0
59.1	16.0	42.0
	14.0	37.0
	12.0	32.0
	10.0	27.0
	8.0	22.0
	6.0	16.0

Model	**RS2000-48-18**
Mounting Hole Type	1/4-20
Mounting Hole Pattern	1 in. grid
Length	8 ft.
Width	4 ft.
Thickness	18 in.
Working Surface	400 series ferromagnetic stainless steel
Surface Flatness	±0.1 mm
Deflection Under Load	$<5.0 \times 10^{-5}$ in. ($<1.3 \times 10^{-3}$ mm)
Maximum Dynamic Deflection Coefficient	0.8×10^{-3}
Core Design	Trussed Honeycomb, Vertically Bonded Closed Cell Construction, 0.010 in. Steel sheet materials, 0.030 in. triple core interface
Broadband Damping	Constrained layer core, damped working surface and composite edge finish
Hole/Core Sealing	Easy clean conical cup 0.75 in. (19 mm) deep, Non-corrosive high impact polymer material
Top and Bottom Skins	3/16 in. (4.8 mm) thick with integrated damping layer


STATIC DEFLECTION (in./lb.)
STATIC DEFLECTION (mm/N)
TABLE THICKNESS, IN. (mm) =8 (203)
=12 (305)
=18 (457)
=24 (610)
=36 (914)
5 FT WIDTH FACTOR = 0.8
TABLE LENGTH
6 1.8
8 2.4
10 3.0
12 3.6
14 4.2
16 4.8
18 5.4
20 ft. 6.0 m.


COMPLIANCE (in./lb.)
COMPLIANCE (mm/N)
TABLE THICKNESS, IN. (mm) =8 (203)
=12 (305)
=18 (457)
=24 (610)
=36 (914)
TABLE LENGTH
6 1.8
8 2.4
10 3.0
12 3.6
14 4.2
16 4.8
18 5.4
20 ft. 6.0 m.

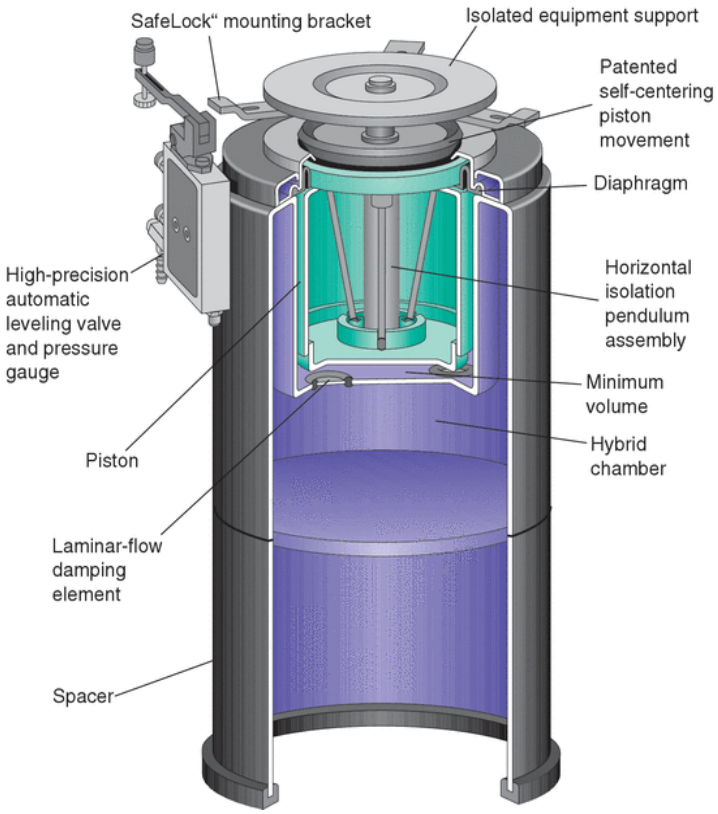

COMPLIANCE (in./lb.)
COMPLIANCE (mm/N)
FREQUENCY (Hz)
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
5
10
100
1000


FREQUENCY (Hz)
300
200
100
70
50
30
TABLE THICKNESS, IN. (mm)
= 36 (914)
= 24 (610)
= 18 (457)
= 12 (305)
= 8 (203)
6
1.8
8
2.4
10
3.0
12
3.6
14
4.2
16
4.8
18
5.4
20 ft.
6.0 m.
TABLE LENGTH

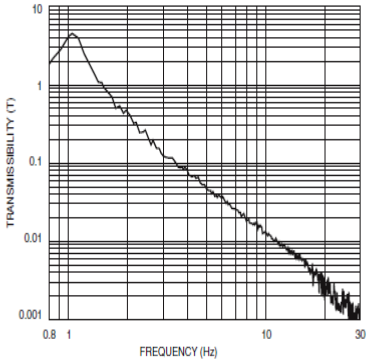
APPENDIX H: Newport Pneumatic Isolators

Specifications

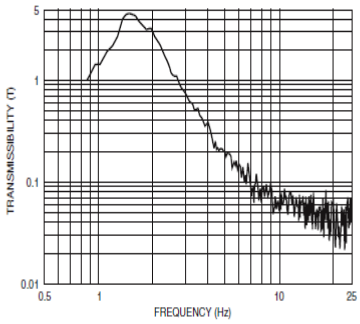
Table Tops:	
Flatness:	±0.005in. (0.13mm)*
Compliance:	Consult your Newport Catalog or Newport directly for the specific compliance and other pertinent table top specifications of your particular table top model.
Isolators:	Stabilizer™ I-2000
Vertical Resonant Frequency:	<1.1 Hz at 80 psi
Horizontal Resonant Frequency:	<1.5 Hz
Recommended Load Range: (per 4 isolators)	660 to 8,000 lb (300 to 3,600 kg)
Automatic Leveling Accuracy:	±0.010 inch (0.25 mm) standard, higher accuracy available on special order
Vertical Adjustment Range:	1.3 inches (33 mm)
Settling Time: (after 5-lb. weight removal)	<1.5 sec.
Typical Air Pressure Range:	10 to 85 psi (0.7 to 6.0 kg/cm^2)


SafeLock" mounting bracket
Isolated equipment support
Patented self-centering piston movement
Diaphragm
High-precision automatic leveling valve and pressure gauge
Horizontal isolation pendulum assembly
Minimum volume
Hybrid chamber
Piston
Laminar-flow damping element
Spacer

Isolation System Transmissibility



Vertical Transmissibility (at maximum recommended load)



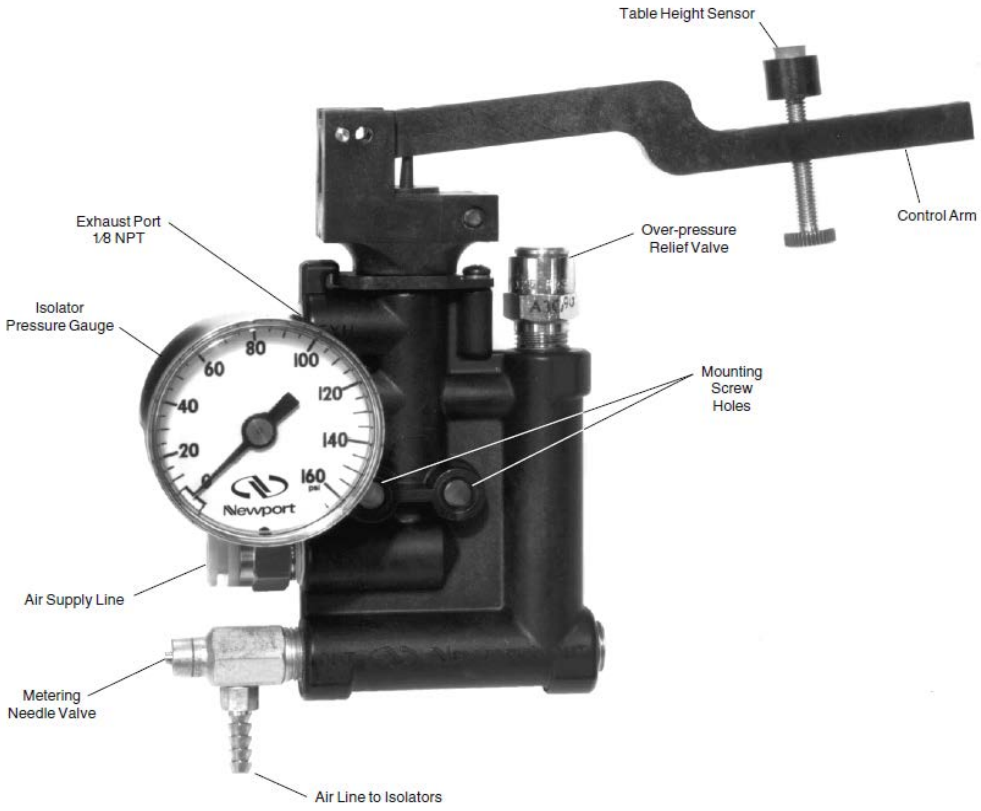
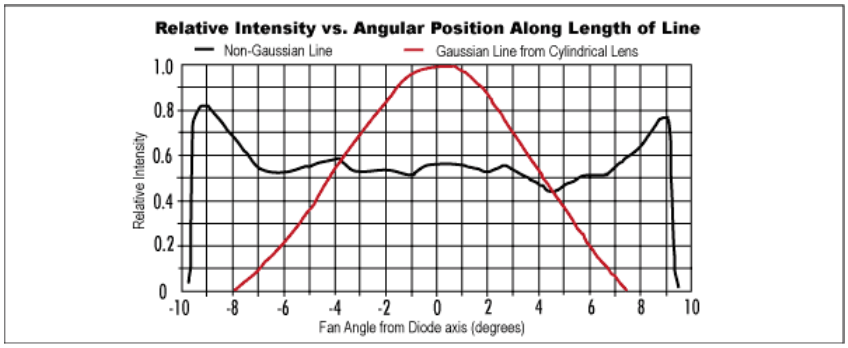


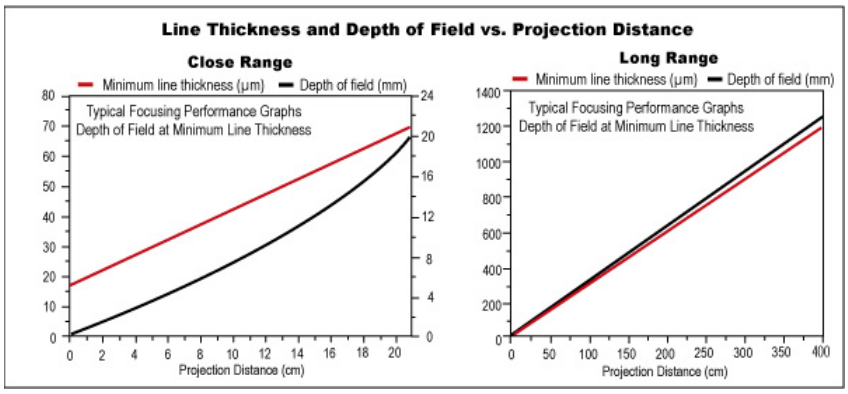
Table Height Sensor
Control Arm
Exhaust Port
1/8 NPT
Over-pressure
Relief Valve
Isolator
Pressure Gauge
Mounting
Screw
Holes
Air Supply Line
Metering
Needle Valve
Air Line to Isolators
Newport

APPENDIX I: Laser Diode

Class	5mW, Class IIIa
Typical Power Output	~75% of max. output power
Beam Diameter	3.8 x 0.9mm, typical collimated beam
Beam Divergence	0.45 x 0.95 mrad, typical collimated beam
Line Width, Focused Spot	<0.001" (25 microns) user adjustable
Focusing Distance	Face of module to past collimation
Dimensions	
Module only	0.750" Diameter +0/-0.005"
Projection Head	0.734" Diameter
Bore Sighting (Beam vs. Housing Alignment)	<3 mrad, collimated beam
Temperature Range	+10°C to +48°C
Frequency Drift	0.25nm / °C
ESD Protection	+8,000 volts
Diode MTBF, calc.	50 - 100,000 hrs, varies with model
Current Draw	65 - 150 mA, varies with model
Input Voltage	5 - 6V DC
Weight	~65 grams
Housing Material	Black Anodized Aluminum

*Class IIIb Models	CDRH certified with key box





APPENDIX J: Experiment Run MATLAB Script

```
%% M-file to Run FFD_BC_3
%*******************************
%LAST USED: 07-22-2011
%  STATUS: Functional=X
%   FB:
%       FSMA: PI                X
%             LMS
%       FSMB: PI
%             LMS
%   FFD:
%       FSMA: PI: PSD           X
%                 ARS          ~X (need to tune better)
%             LMS: PSD
%                  ARS
%       FSMB: PI: PSD
%                 ARS
%             LMS: PSD
%                  ARS
%    Target Tracking X OT5 FB  X (may want to tune better)
%    Target Tracking Y OT5 FB  X (may want to tune better)
 %   Target Tracking X OT4
%    Target Tracking Y
%
%% Use this for saving data (May need to update variables!)
%*******************************
% Check to make sure date for save file is correct!
%*******************************
%for ii = 1:4;  %using this for running the same experiment numerous times
%ptime = input('input pause time in sec   ');
%ptime = 0; %This is a pause time
% pltfrf = 0;         %   Set to one to run FRF
% if isempty(ptime)
%     ptime=0;
% end
%% Basic Run Parameters
calibrate=1;            %Set to '1' to use Calibration Constants
Ts=0.001;               %sample time
Fs=1/Ts;                %sample Freguency

fintime = 17;            %Length of data run
MeanOff = 0;            % subtract running mean for HINF
                        % 1 = on
%-----------------------------------------
% Actuator input (sinusoid, max 4 signals)
%   amp in volts, freq in Hz
%-----------------------------------------
sfreq = 10;             %Actuator First Frequency
aa = 0;                 %Make '1' to Run all Frequncies for Actuator 1
bb = 0;                 %Make '1' to Run all Frequencies for Actuator 2
%-----------------------------------------
%Actuator 1 (Pitch/Yaw)
shakeramp = [3    2       0       0];
```

```
shakerfreq= [12       38         37         43]; %[sfreq       13         27
47];
shakerampb = [3    3     0       0];
shakerfreqb= [9       46         27         47]; %[sfreq       13         27
47];
shaker_switch=7; % Shaker switch time for frequencies
% shakerfreq= [sfreq       16         28         47]; %Moran 10Nov
shakerphase = [0   0   0   0];
shaker_start=1;         %start time of vibrations in secs
shaker_end=fintime;
noise_power=0.02*aa;  %noise power for Band Limited White Noise
                      %(usually use about 0.01)
%-----------------------------------------
%Actuator 2 (Roll)
shakeramp2 = [3*0  2*0      1*bb       1*bb];
shakerfreq2= [10      23         41       51]; %[17      23         41
51];
shakerphase2 = [pi/4*1   pi/3*0   0   0];
shaker_start2=shaker_start;         %start time of vibrations in secs
shaker_end2=fintime;
noise_power2=0.02*bb;  %noise power for Band Limited White Noise
                       %(usually use about 0.01)
%-----------------------------------------
%Chirp Parameters (set chirp_on to 1 for chirp signal, 0 to input freq)
chirp_on = 0; IA_chirp_gain=0.8;
IA_init_freq = 1;  IA_final_freq = 150; IA_targ_time = 101;  %Chirp
Parameters
%-----------------------------------------
%Distance from Last FSM face to Target
dist_targ = 4.4967; %m
%dist_targ = 8; %m
%Distance from Laser Source to FSM

%dist_FSM = (0.365+0.427)*1.00; %m
dist_FSM = 0;
% dist_FSM = 0.2275*0; %m
%-----------------------------------------
% Distance from Plate's "Center of Rotation" to FSM
    w = 0.0635; %m       (originally 0.0635)
    h = 0.1175; %m   (originally 0.1175)
    d= 0.3048; %m        (originally 0.3048)
%-----------------------------------------
FSM_position = [0,0,0];
%-----------------------------------------
Run_Mean = 0;       %Set to '1' to Subtract Running Mean from Rate Sensor
Data
a_Run_Mean = 0;     %Set to '1' to Subtract Running Mean from Accelerometer
Data
Filter_Mean = 1;    %Set to '1' to Subtract the jitter free "drift" signal
from ARS data

% Target tracking curve fit parameters

%X
%Target tracking slope x
```

```
Mx = -0.9212; %(curve fit=-0.9212)
%Target tracking offset x
Bx = -5.11; %(curve fit = -3.714)

%Y
%Target tracking slope y
My = 0; %(curve fit=)
%Target tracking offset y
By = 0; %(curve fit = )



targtrackxprop=0.044;
targtrackxint=.81;
targtrackxderiv=0;

targtrackyprop=0.044;
targtrackyint=.81;
targtrackyderiv=0;

targtrackxprop2=.26;
targtrackxint2=4.6;
targtrackxderiv2=0;

targtrackyprop2=.26;
targtrackyint2=4.6;
targtrackyderiv2=0;


%-----------------------------------------
trigger=2;  %Trigger for Beam Profile, 1=Trigger ON, 2=Trigger OFF

%% Plot Selection
%   1 = plot, 0 = don't plot
%-----------------------------------------
%                       Title of plot
acc          =0;    %   Accelerometer Output
OT_plot      =0;    %   OT1,OT2 and OT3 PSD position in \mu','m and FSM pos
in volts
volt_fig     =1;    %   FSM position in mrad
OT3_pos      =0;    %   OT3 pos on detector with OT3 x and y vs. time
OT1_pos      =0;    %   OT1 pos on detector vs. time
OT2_pos      =0;    %   OT2 pos on detector vs. time
OT4_pos      =0;    %   OT4 pos on detector vs. time
OT5_pos      =1;    %   OT5 pos on detector vs. time
OT6_pos      =0;    %   OT6 pos on detector vs. time
OT_plot_compare=0;  %   OT5 and prediction vs. time
percent_imp  =1;    %   Percent improvement in Target position
powerplot    =0;    %   RMS laser power
fsm_cmd      =1;    %   FSM command voltage
psd_plt_x    =0;    %   Periodogram of accels/displacement x dir
psd_plt_y    =1;    %   Periodogram of accels/displacement x/y dir
psd_plt_z    =0;    %   Periodogram of accels/displacement z dir
```

```
test_inp     = 0;   %   Set to one to plot test input signal
stats        = 1;   %   Set to calculate statistics for output
rot          = 0;   %   Set to plot rotation rates
rot_cal      = 0;   %   Set to plot calibrated rotation rates
rotations    = 0;   %   Set to plot Rotations from PSDs and Rate Sensors
jitter       = 0;   %   Set to plot jitter angle
temp         =0;   %   Set to plot temperatures

%% Control Selection:
%-----------------------------------------
% Select Rotations from either PSDs or Rate Sensors for use with Control
    % 1 = PSD Calc
    % 2 = Rate Sensors (Integration Only)
    % 3 = Rate Sensors with Prediction Algorithm
    % 4 = Rate Sensor with Accels and Prediction Algorithm
    PSD_or_Rate_Sensors =1;

%-----------------------------------------
%   Select Target Position Control or Required FSM Theta Control
       % 1 = Tgt Position with PI
       % 2 = Req Theta
    Tgt_Pos_or_Req_Theta = 1;

%-----------------------------------------
%   Select FeedBack or FeedForward Control for use with Target Position
%       Control Above (1 must be selected above)
%    1 = FeedBack;  2 = FeedForward
     Back_or_Forward = 1;
     if (Back_or_Forward == 2)
         PredFilter = 0;
     else
         PredFilter = 0;
     end
        A_x_ffd_sel = Back_or_Forward; %(x axis at tgt)
        A_y_ffd_sel = Back_or_Forward; %(y axis at tgt)
%-----------------------------------------
%   Select Target tracking source
    %1=OFF(Beacon Laser on OT4)   2=ON (OT5 feedback)
    OT5FBX = 2;
    OT5FBY = 2;
   trackstart=1.5; %delay before tracking starts

%% Control Parameters:
%-----------------------------------------
%   PID gains for PI Controller (Control A)
    %(Kcr_x=0.0158, Pcr_x=0.002 and y crit gain = 0.031)

if Back_or_Forward == 1;
    %FB ideal gains:
    fsm1px = 0.04*0.45*1.0;
    fsm1ix = fsm1px*1.2/0.001;
    fsm1dx = 0;
    fsm1py = 0.1*0.45*0.5;
```

```
    fsm1iy = fsm1py*1.2/0.001;
    fsm1dy = 0;
 elseif Back_or_Forward == 2;
    if PSD_or_Rate_Sensors == 1;
    %PSM ideal gains:
    fsm1px = 0.03*0.45;
    fsm1ix = fsm1px*1.4/0.001;
    fsm1dx = 0;
    fsm1py = 0.04*0.45;
    fsm1iy = fsm1py*1.5/0.001;
    fsm1dy = 0;
    else
    %ARS ideal                    PSM ideal                  FB ideal
    fsm1px = 0.007*0.45*2.5;        %0.03*0.45                %0.04*0.45
    fsm1ix = fsm1px*1.9/0.001;   %fsm1px*1.4/0.001       %fsm1px*1.2/0.001
    fsm1dx = 0;
    fsm1py = 0.005*0.45*1;         %0.04*0.45              %0.1*0.45
    fsm1iy = fsm1py*3.1/0.001;   %fsm1py*1.5/0.001       %fsm1py*1.2/0.001
    fsm1dy = 0;
    end
end

%     fsm1px = 0.0079*on_off; %0.016*0.45*za*1.1;   %0.04*0.45*za
%     fsm1ix = 11.4048;
%     fsm1dx = 0;                %0
%     fsm1py = 0.031*0.45*on_off;     %0.1*0.45*za
%     fsm1iy = fsm1py*1.2/0.001*on_off;
%     fsm1dy = 0;

    % Use these for tuning the PI Controller
    PI_tune_step_value= 0; %step value
    x_PI_tune = 3; %time for x axis step
    y_PI_tune = 3; %time for y axis step
%-----------------------------------------
%   LMS parameters for LMS Controller (Control B)
%     mux=0.012;      leakx=1;      % x axis adaption rate and leakage factor
%     muy=0.020;      leaky=1;      % y axis adaption rate and leakage factor
    mux=0.007;      leakx=0.998;      % x axis adaption rate and leakage
factor
    muy=0.013;      leaky=0.998;      % y axis adaption rate and leakage
factor
    w0x = 0;        w0y = 0;        % initial tap gains
    biasx=-0.005*1;   biasy=0.05*1;     % estimate of bias correction
    ax_to_mx=1;       ay_to_my=14;   % estimate of gain correction for FSM to
accel
    ot2y_to_m2y = -1/10;
    mu_y_error = 0.05; leak_y_error = 1.0;
    adapt_y_error = 0.0;
    mu_x_error = 0.05; leak_x_error = 1.0;
%-----------------------------------------
%   Reference Signal Selection
%   1=OT-1, 2=Accel-2 (a2x and a2y), 3 = rate sensor (pitch, roll)
    x_ref_sel=1;    y_ref_sel=1;
    zz=1;  % number of delays for the predictor ref signal
%-----------------------------------------
```

```
%   Error source selection
%   1=mirror postion, 2=OT3 position, 3=OT2 position
    x_error_sel=2;  y_error_sel=2;
    accel_lag = 1.05;
    OT2y_lag = 1;
%----------------------------------------
% parallel controllers cmd - 1=single, 2 = parallel A and B
    par_cntlrsA = 1; par_cntlrsB = 1;
%----------------------------------------
%   Test Parameters for sinusoid (max 4 signals)
%   amplitude in Volts, frequency in Hz
    x_test_amp= [0.02        0     0       0];
    x_test_freq=[2          0       0       0];
    y_test_amp= [0.02        0       0       0];
    y_test_freq=[1          0       0       0];
%   time in sec, value in mrad (max = 13.1 mrad)
    y_step_time = 1;
    y_step_value = 0.1; y_step_value = y_step_value*10/26.2; %convert to
volts
    x_step_time = 1;
    x_step_value = 0.1; x_step_value = x_step_value*10/26.2;
    imp_delay = 1;  imp_delay=round(imp_delay/Ts);  %delay time to impulse in
sec
    imp_mag = -0.3;  imp_mag = imp_mag*10/26.2+0.03*0;  % impulse mag in mrad
    init_freq = 1;  final_freq = 1000; targ_time = 120;  %Chirp Parameters
    chirp_gain = 0.262;  chirp_gain=chirp_gain*10/26.2;
    stepOTxstart = 1;
    stepOTystart = 1;

% FSM_Acal_x = 2.62*1.3;  FSM_Acal_y = 2.62*1.40;
FSM_Acal_x = 2.62*1;  FSM_Acal_y = 2.62*1;

%% Plot Parameters
%----------------------------------------
    plot_time=1.5; %length of plot in seconds
    delay_time=shaker_start+0.5;  %delay before start of example plot
    adapt=0.5+delay_time+plot_time; %modify adaption to be after delay
    x_plot_bias=350;    y_plot_bias=750; %amt to bias example signal
    pbiasy = 900; pbiasx = 300;
    pidstart = adapt; % PID control start, sec, before adaption
    req_theta_start=pidstart; %Required Theta Control start
    %controlstart = pidstart;  %Start HINF Control
    controlstart = 1; % Added 21 Oct by ROB

%     xzero = 0;  yzero = 0.0;
%     xzero2 = 0.8;  yzero2 = 0.8;
%% Run Model
%----------------------------------------
%pause(ptime);
set_param('FFD_HINF_1', 'SimulationCommand', 'update')
    tg.StopTime=fintime;
tg = xpctarget.xpc
+tg
pause(fintime+0.5);
-tg
```

```
clear tt oo
tt=tg.TimeLog;

%% Get Vairables for Plotting and Analysis:

% FSM position in volts
    m1x=tg.OutputLog(:,46);
    m1y=tg.OutputLog(:,47);
    m2x=tg.OutputLog(:,48);
    m2y=tg.OutputLog(:,49);
    %----------------------------------------
% OnTrak position in micrometers
%   Corrected to Platform coordinate system
%   (except OT2x and OT3x are +/-z direction)
%   Negative value for OT mounted on platform
%   since upward motion results in downward
%   displacement
    ot1x=-500*tg.OutputLog(:,3);   %pt1 x
    ot1y=-500*tg.OutputLog(:,4);   %pt1 y
    ot2x=500*tg.OutputLog(:,5);    %pt2 x
    ot2y=-500*tg.OutputLog(:,6);   %pt2 y
    ot3x=-500*tg.OutputLog(:,7);  %pt3 x
    ot3y=-500*tg.OutputLog(:,8);   %pt3 y
    ot4x=500*tg.OutputLog(:,9);   %pt1 x'
    ot4y=-500*tg.OutputLog(:,10);  %pt1 z
%     ot5x=-500*tg.OutputLog(:,11);  % Target x pos
    ot5x=500*tg.OutputLog(:,11);  % Target x pos
    ot5y=500*tg.OutputLog(:,12);   % Target y pos
    ot6x = 500*tg.OutputLog(:,61);
    ot6y = 500*tg.OutputLog(:,62);
%     ot5x_store(ii) = mean(ot5x((1/Ts+100):end));
%     ot5y_store(ii) = mean(ot5y((1/Ts+100):end));
%     %ot6x=500*tg.OutputLog(:,31);   %pt2 z'
%     %ot6y=500*tg.OutputLog(:,33);   %pt2 x
%     %ot7x=-500*tg.OutputLog(:,34);  %pt3 z'
%     %ot7y=-500*tg.OutputLog(:,35);  %pt3 x
%     E1=tg.OutputLog(:,24);  %was 34
%     E2=tg.OutputLog(:,35);
%         c0=5.306462;
%         c1=-25.30863;
%         c2=-0.777941;
%         c3=-0.507258;
%         temp1= c0 + c1.*log(E1) + c2.*(log(E1)).^2 + c3.*(log(E1)).^3;
%         temp2= c0 + c1.*log(E2) + c2.*(log(E2)).^2 + c3.*(log(E2)).^3;
%----------------------------------------
% Angular Rate Sensors
    pitch_rate=tg.OutputLog(:,16)./kp.*1e6;
    roll_rate=tg.OutputLog(:,17)./kr.*1e6;
    yaw_rate=tg.OutputLog(:,18)./ky.*1e6;
    pitch_rate_calibrated=tg.OutputLog(:,19);
    roll_rate_calibrated=tg.OutputLog(:,20);
    yaw_rate_calibrated=tg.OutputLog(:,21);
    %PSD Calculated angles
    pitch_1=tg.OutputLog(:,13);
    roll_1=tg.OutputLog(:,14);
```

```
    yaw_1=tg.OutputLog(:,15);
    %ARS integrated angles
    pitch_2=tg.OutputLog(:,22);
    roll_2=tg.OutputLog(:,23);
    yaw_2=tg.OutputLog(:,24);
    %ARS predicted angles
    pitch_3=tg.OutputLog(:,25);
    roll_3=tg.OutputLog(:,26);
    yaw_3=tg.OutputLog(:,27);
%Accelerometers
    % Accelerometer predicted angles
%     a_pitch_accel=tg.OutputLog(:,67);
%     a_roll_accel=tg.OutputLog(:,68);
%     a_yaw_accel=tg.OutputLog(:,69);
%     a_pitch_rate=tg.OutputLog(:,64);
%     a_roll_rate=tg.OutputLog(:,65);
%     a_yaw_rate=tg.OutputLog(:,66);
%     a_pitch_angle=tg.OutputLog(:,61);
%     a_roll_angle=tg.OutputLog(:,62);
%     a_yaw_angle=tg.OutputLog(:,63);
%ARS+Accelerometer Angle Calculation Block
%     ARSaccel_pitch_angle=tg.OutputLog(:,61);
%     ARSaccel_roll_angle=tg.OutputLog(:,62);
%     ARSaccel_yaw_angle=tg.OutputLog(:,63);
%-----------------------------------------
% Position at Target from Beam Prediction Algorithm
    x_pred = -500*tg.OutputLog(:,50); %yes, the minus sign is supposed ot be
there
    y_pred = 500*tg.OutputLog(:,51);
%     x_pred_store(ii) = mean(x_pred((1/Ts+100):end));
%     y_pred_store(ii) = mean(y_pred((1/Ts+100):end));
% Position at Target from Prof. Radice's Formula
%     x_pos_radice=500*tg.OutputLog(:,50);
%     y_pos_radice=500*tg.OutputLog(:,51);
%
%end
%% Calculate Jitter at Target
%-----------------------------------------
shake=find(tt>=shaker_start);
shake=shake(1);
control=find(tt>=pidstart);
control=control(1)-1;
done=length(tt);

ot5r = ((ot5y.^2+ot5x.^2).^0.5); % miss dist in um
ot5j = ot5r./dist_targ;  % jitter in urad

%Use this if using Control with Mirror Command b/c it does not target (0,0)
if Tgt_Pos_or_Req_Theta == 2;
    ot5r = (((ot5y-mean(ot5y(1:shake))).^2 + (ot5x-
mean(ot5x(1:shake))).^2).^0.5);
    ot5j = ot5r./dist_targ;
end

% Mean Jitter Angle Before and after Control
```

```
Mean_jitter_before=mean(ot5j(shake:control));
Mean_jitter_after=mean(ot5j(control:done));
imp_mean_jitter=((Mean_jitter_before-
Mean_jitter_after)/Mean_jitter_before)*100;

%Running mean of mean jitter angle
rmean_jitter=smooth(ot5j,5,'moving');

% % Jitter in Calculated Signal
% if Tgt_Pos_or_Req_Theta == 1;
    ot5x_calc = x_pred;
    ot5y_calc = y_pred;
% end
% if Tgt_Pos_or_Req_Theta == 2;
%     ot5x_calc = x_pos_radice;
%     ot5y_calc = y_pos_radice;
% end
ot5r_calc=((ot5y_calc.^2 + ot5x_calc.^2).^.5);
ot5j_calc=ot5r_calc./dist_targ;
jitter_error=ot5j_calc-ot5j;
RMS_jitter_error=sum(sqrt(jitter_error(shake:done).^2))/length(jitter_error(s
hake:done));
% Percent Improvement with Control
%ot5j_shake=mean(ot5j(shake:control,:));
%ot5j_control=mean(ot5j(control:done,:));
%jstdin=sqrt(var(ot5j(shake:control)));
%jstdout=sqrt(var(ot5j(control:done)));
%ystdin=sqrt(var(ot5y(shake:control)));
%xstdin=sqrt(var(ot5x(shake:control)));
%ystdout=sqrt(var(ot5y(control:done)));
%xstdout=sqrt(var(ot5x(control:done)));
%impj=(1-(jstdout/jstdin))*100;
%impy=(1-(ystdout/ystdin))*100;
%impx=(1-(xstdout/xstdin))*100;

%%   Acceleration in volts
Ox = tg.OutputLog(:,28);
Oy = tg.OutputLog(:,29);
Oz = tg.OutputLog(:,30);
Ay = tg.OutputLog(:,31);
Az = tg.OutputLog(:,32);
Bx = tg.OutputLog(:,33);
Bz = tg.OutputLog(:,34);
Cx = tg.OutputLog(:,35);
Cy = tg.OutputLog(:,36);

%----------------------------------------
%   Test output in volts.
    xtest=tg.OutputLog(:,1);
    ytest=tg.OutputLog(:,2);
%----------------------------------------
%   FSM command voltage and Misc inputs/outputs

%       FSM commanded positions(volts)
```

```
        m1xc=tg.OutputLog(:,46);
        m1yc=tg.OutputLog(:,47);
        m2xc=tg.OutputLog(:,48);
        m2yc=tg.OutputLog(:,49);
%       FSM actual positions (volts)
        FSMAVx=tg.OutputLog(:,40);
        FSMAVy=tg.OutputLog(:,41);
        FSMBVx=tg.OutputLog(:,42);
        FSMBVy=tg.OutputLog(:,43);

        IA_1_input=tg.OutputLog(:,44);
        IA_2_input=tg.OutputLog(:,45);
%
%         req_theta_x=tg.OutputLog(:,52);
%         req_theta_y=tg.OutputLog(:,53);

%-------------------------------------------
%       Target Intercept, Non-Rotated
%-------------------------------------------

    tnr_x = tg.OutputLog(:,58);
    tnr_y = tg.OutputLog(:,59);
    tnr_z = tg.OutputLog(:,60);


%% Calculation of plot samples
    frt1=size(tg.OutputLog);
    calc_sample=frt1(1,1)-(round(plot_time/Ts+0.1/Ts));  % start of plot
    start_sample=round(0.5/Ts);
    plot_sample=calc_sample+round(plot_time/Ts);         % end of plot
    exsamp=round(delay_time/Ts);
exsamp_end=round(plot_time/Ts+delay_time/Ts);
    tt1=tt(calc_sample:plot_sample);
    ot5xplt=ot5x(calc_sample:plot_sample);
    ot5yplt=ot5y(calc_sample:plot_sample);
    ot5xex=ot5x(exsamp:exsamp_end); ot5xex=ot5xex-mean(ot5xex)+x_plot_bias;
    ot5yex=ot5y(exsamp:exsamp_end); ot5yex=ot5yex-mean(ot5yex)+y_plot_bias;
    title_ctr='LMS';
    title_error='OT3';

%%   Statistics
if stats==1;
    ystdin=sqrt(var(ot5yex));               %standard deviation of input
    xstdin=sqrt(var(ot5xex));

    ystd=sqrt(var(ot5yplt));                %standard deviation of output
    xstd=sqrt(var(ot5xplt));

    impx=(1-(xstd/xstdin))*100;             %percent improvement in st.dev.
    impy=(1-(ystd/ystdin))*100;

    meanx=round(1000*mean(ot5xplt));        %mean position at OT3 in
nanometers
```

```
    meany=round(1000*mean(ot5yplt));

%     mean_start_power =
round(1000*mean(laser_power(start_sample:round((shaker_start/Ts-10)))));
%     mean_dist_power =
round(1000*mean(laser_power((exsamp+round(0.25/Ts)):exsamp_end)));
%     mean_rec_power =
round(1000*mean(laser_power(calc_sample:plot_sample)));
    ot5ym = ot5y-mean(ot5y);
    ot5xm = ot5x-mean(ot5x);

%     rot_error_pitch= (mean(pitch_1(1000:6000)))-
mean(pitch_3(1000:6000))/mean(pitch_1(1000:6000)); %error between PSD and ARS
platform rotation calcs-referenced to PSD
%     rot_error_roll=(mean(roll_1(1000:6000))-
mean(roll_3(1000:6000)))/mean(roll_1(1000:6000));
%     rot_error_yaw=(mean(yaw_1(1000:6000))-
mean(yaw_3(1000:6000)))/mean(yaw_1(1000:6000));
rot_error_pitch=sqrt(mean((pitch_1(1001:end)-pitch_3(1001:end)).^2));
rot_error_roll=sqrt(mean((roll_1(1001:end)-roll_3(1001:end)).^2));
rot_error_yaw=sqrt(mean((yaw_1(1001:end)-yaw_3(1001:end)).^2));
end

%% Calculate Frequency Spectrum
if psd_plt_x==1;
%     [Pa1x,ff]=periodogram(a1x,nfft,'onesided',window,Fs);
Pa1x=10*log10(Pa1x);
%     [Pa1y,ff]=periodogram(a1y,nfft,'onesided',window,Fs);
Pa1y=10*log10(Pa1y);
%     [Pa1z,ff]=periodogram(a1z,nfft,'onesided',window,Fs);
Pa1z=10*log10(Pa1z);
%     [Pa2x,ff]=periodogram(a2x,nfft,'onesided',window,Fs);
Pa2x=10*log10(Pa2x);
%     [Pa2y,ff]=periodogram(a2y,nfft,'onesided',window,Fs);
Pa2y=10*log10(Pa2y);
%     [Pa2z,ff]=periodogram(a2z,nfft,'onesided',window,Fs);
%     Pa2z=10*log10(Pa2z);

[Pot1x,ff]=periodogram(ot1x,nfft,'onesided',window,Fs);Pot1x=10*log10(Pot1x);

[Pot1y,ff]=periodogram(ot1y,nfft,'onesided',window,Fs);Pot1y=10*log10(Pot1y);

[Pot3x,ff]=periodogram(ot3x,nfft,'onesided',window,Fs);Pot3x=10*log10(Pot3x);

[Pot3y,ff]=periodogram(ot3y,nfft,'onesided',window,Fs);Pot3y=10*log10(Pot3y);

[Pot2x,ff]=periodogram(ot2x,nfft,'onesided',window,Fs);Pot2x=10*log10(Pot2x);

[Pot2y,ff]=periodogram(ot2y,nfft,'onesided',window,Fs);Pot2y=10*log10(Pot2y);

[Pot5x,ff]=periodogram(ot5x,nfft,'onesided',window,Fs);Pot5x=10*log10(Pot5x);

[Pot5y,ff]=periodogram(ot5y,nfft,'onesided',window,Fs);Pot5y=10*log10(Pot5y);
end
```

```
%% Plots:
%-------------------------------------------------------------------------
if volt_fig==1
%     vmeanstx = roundn(mean(m1x(round(0.9/Ts):round(1.0/Ts))),-2);
%     vmeansty = roundn(mean(m1y(round(0.9/Ts):round(1.0/Ts))),-2);
%     vmeanspx = roundn(mean(m1x((round(1.4/Ts)):round(1.6/Ts))),-2);
%     vmeanspy = roundn(mean(m1y((round(1.4/Ts)):round(1.6/Ts))),-2);
%     vmeanstx2 = roundn(mean(m2x(round(0.9/Ts):round(1.0/Ts))),-2);
%     vmeansty2 = roundn(mean(m2y(round(0.9/Ts):round(1.0/Ts))),-2);
%     vmeanspx2 = roundn(mean(m2x((round(1.4/Ts)):round(1.6/Ts))),-2);
%     vmeanspy2 = roundn(mean(m2y((round(1.4/Ts)):round(1.6/Ts))),-2);

figure(2)
    m1xp = FSMBVx.*1; m1yp = FSMBVy.*1;
    subplot(2,1,1)
    plot(tt,(m1xp-xzero),tt,m2xc*2.62),grid,zoom,legend('Act','Cmd')
    ylabel('fsm x pos, mrad')
    title('FSM pos vs. time')
    subplot(2,1,2)
    plot(tt,(m1yp-yzero),tt,m2yc*2.62),grid,zoom,legend('Act','Cmd')
    ylabel('fsm y pos, mrad')
    xlabel('time,sec')
end
%-------------------------------------------------------------------------
if OT_plot==1;
%figure('Name','OT Plot','NumberTitle','on')
figure(3)
    subplot(2,1,1)
    plot(tt,(ot1y-mean(ot1y)),tt,(ot2y-mean(ot2y)),tt,(ot3y-mean(ot3y)),...
        tt,(ot4y-mean(ot4y)),tt,(ot5y-mean(ot5y))),grid,zoom
    legend('ot1y','ot2y','ot3y','ot4y','ot5y')
    ylabel(['\mu','m '])
    title(['OT1,OT2,OT3 and OT4 PSD position in \mu','m '])
    subplot(2,1,2)
    plot(tt,(ot1x-mean(ot1x)),tt,(ot2x-mean(ot2x)),tt,(ot3x-mean(ot3x)),...
        tt,(ot4x-mean(ot4x)),tt,(ot5x-mean(ot5x))),grid,zoom
    legend('ot1x','ot2x','ot3x','ot4x','ot5x')
    %axis([tt(calc_sample) tt(plot_sample) -inf inf ])
    %axis([tt(calc_sample) tt(plot_sample) -50 250 ])
    ylabel(['\mu','m '])
    xlabel('time,sec')
end
%-------------------------------------------------------------------------
if OT3_pos==1;
%figure('Name','OT 3','NumberTitle','on')
figure(4)
    subplot(2,1,1)
    plot(tt,ot3x),grid,zoom
     ylabel(['x pos, \mu','m'])
     title('OT3 pos vs. time')

     subplot(2,1,2)
     plot(tt,ot3y),grid,zoom
     ylabel(['y pos, \mu','m'])
```

```
     xlabel('time,sec')
end
%-------------------------------------------------------------------------
% if acc==1
% %figure('Name','Accels','NumberTitle','on')
% figure(5)
%     subplot(3,1,1)
%     plot(tt,(a1x-mean(a1x)),tt,(a2x-mean(a2x)),tt,(a3x-
mean(a3x))),grid,zoom
%     xlabel('time,sec')
%     ylabel('Acceleration, g_x')
%     legend('A1x','A2x','A3x')
%     title('Accelerometer Output')
%     subplot(3,1,2)
%     plot(tt,(a1y-mean(a1y)),tt,(a2y-mean(a2y)),tt,(a3y-
mean(a3y))),grid,zoom
%     xlabel('time,sec')
%     ylabel('Acceleration, g_y')
%     legend('A1y','A2y','A3y')
%     %title('Accelerometer Output')
%     subplot(3,1,3)
%     plot(tt,(a1z-mean(a1z)),tt,(a2z-mean(a2z)),tt,(a3z-
mean(a3z))),grid,zoom
%     xlabel('time,sec')
%     ylabel('Acceleration, g_z')
%     legend('A1z','A2z','A3z')
%     title('Accelerometer Output')
% end
%-------------------------------------------------------------------------
if percent_imp==1;
%figure('Name','% imp','NumberTitle','on')
figure(6)
subplot(2,1,1)
    plot(tt1,ot5yex,tt1,ot5yplt),grid,zoom
    title(['36 stage ',char(title_ctr),' Controller: Improvement: X  ',...
            num2str(impx),' %, Y  ',num2str(impy),' %   Mean X :
',num2str(meanx),' nm  Y : ',num2str(meany),...
            ' nm ',char(title_error)])
    %text(0.01,20,' X Axis','FontWeight','bold')
    %title('35 Hz vibration signal - amplitude 1.3 V')
    legend('input jitter','controlled beam')
    ylabel(['y-pos, \mu','m'])
    %axis([tt(calc_sample) tt(plot_sample) -inf inf ])
    axis([tt(calc_sample) tt(plot_sample) -100 y_plot_bias+pbiasy ])
    subplot(2,1,2)
    plot(tt1,ot5xex,tt1,ot5xplt),grid,zoom
    title(['Std Dev of error: X Axis input - ',num2str(xstdin),'\mu, Output -
',num2str(xstd),...
            '\mu; Y Axis input - ',num2str(ystdin),'\mu, Output -
',num2str(ystd),'\mu'])
    %text(0.01,20,' Y Axis','FontWeight','bold')
    legend('input jitter','controlled beam')
    %axis([tt(calc_sample) tt(plot_sample) -inf inf ])
    axis([tt(calc_sample) tt(plot_sample) -100 x_plot_bias+pbiasx ])
    ylabel(['x pos, \mu','m'])
```

```
    xlabel('time,sec')
end
%-------------------------------------------------------------------------
if powerplot==1;
%figure('Name','Laser Pwr','NumberTitle','on')
figure(7)
    ystart=mean_start_power-0.10*mean_start_power;
    yend=mean_start_power+0.02*mean_start_power;
    laser_smooth=smooth(laser_power,150);
    plot(tt,laser_smooth*1000),grid,zoom
    %plot(tt,laser_power*1000),grid,zoom
    axis([-inf inf  ystart yend ])
    title(['Laser power - ','starting power: ',num2str(mean_start_power),...
        '\mu W; disturbed power: ',num2str(mean_dist_power),'\mu W;
recovered power: '...
        num2str(mean_rec_power),'\mu W (using a 150 pt moving avg filter)'])
    ylabel(['power, \mu','W'])
    xlabel('time,secs')
    line([shaker_start shaker_start],[ystart yend],'color','r')
    line([adapt adapt],[ystart yend],'color','g')
    text(shaker_start,yend-5,['Start vibration
\rightarrow'],'HorizontalAlignment','right','VerticalAlignment','bottom','Fon
tWeight','bold')
    text(adapt,yend-5,['\leftarrow Controller start
'],'HorizontalAlignment','left','VerticalAlignment','bottom','FontWeight','bo
ld')
end
%-------------------------------------------------------------------------
if fsm_cmd==1;
%figure('Name','FSM Cmd','NumberTitle','on')
figure(8)
    subplot(2,1,1)
        plot(tt,m1xc,tt,m1yc),grid,zoom
        legend('FSMAcmdx','FSMAcmdy')
        xlabel('time,secs')
        ylabel('FSMA cmd, volts')
        title('FSMA command voltage')
    subplot(2,1,2)
        plot(tt,m2xc,tt,m2yc),grid,zoom
        legend('FSMBx','FSMBy')
        xlabel('time,secs')
        ylabel('FSMB cmd, volts')
        title('FSMB command voltage')
end
%-------------------------------------------------------------------------
if psd_plt_x==1;
%figure('Name','Accelx OTx PSD','NumberTitle','on')
figure(9)
    subplot(2,1,1)
        plot(ff,Pa1x,ff,Pa2x),grid,zoom
        title('Spectral Density a1x and a2x - Accel. in X direction')
        xlabel('frequency, Hz')
        ylabel('dB/Hz')
        legend('accel 1','accel 2')
    subplot(2,1,2)
```

```
        plot(ff,Pot1x),grid,zoom
        title('Spectral Density OT1x - displacement in X direction')
        xlabel('frequency, Hz')
        ylabel('dB/Hz')
        legend('Source')
end
%--------------------------------------------------------------------
if psd_plt_y==1;
window = 8192/2; nfft=[];      %   Window and size of FFT
%     [Pa2x,ff]=periodogram(a2x,nfft,'onesided',window,Fs);
Pa2x=10*log10(Pa2x);
%     [Pa2y,ff]=periodogram(a2y,nfft,'onesided',window,Fs);
Pa2y=10*log10(Pa2y);
%
[Pot5y,ff]=periodogram(ot5y,nfft,'onesided',window,Fs);Pot5y=10*log10(Pot5y);
%
[Pot5x,ff]=periodogram(ot5x,nfft,'onesided',window,Fs);Pot5x=10*log10(Pot5x);
    noverlap = [];
%     [Pa2x,ff]=pwelch(a2x(5/Ts:end),window,noverlap,nfft,Fs);
Pa2x=10*log10(Pa2x);
%     [Pa2y,ff]=pwelch(a2y(5/Ts:end),window,noverlap,nfft,Fs);
Pa2y=10*log10(Pa2y);
    %  Subtract calibration if FFD
    %if (x_ffd_sel|A_ffd_sel)==1;
    [Pot5y,ff]=pwelch((ot5y(3/Ts:end)-
(cal_tgty*500)),window,noverlap,nfft,Fs);Pot5y=10*log10(Pot5y);
    [Pot5x,ff]=pwelch((ot5x(3/Ts:end)-
(cal_tgtx*500)),window,noverlap,nfft,Fs);Pot5x=10*log10(Pot5x);
%     if h_Pot5x == [];h_Pot5x = Pot5x;end;
%     if h_Pot5y == [];h_Pot5y = Pot5y;end;
    %else
%
[Pot5y,ff]=pwelch((ot5y(5/Ts:end)),window,noverlap,nfft,Fs);Pot5y=10*log10(Po
t5y);
%
[Pot5x,ff]=pwelch((ot5x(5/Ts:end)),window,noverlap,nfft,Fs);Pot5x=10*log10(Po
t5x);
%     end
%figure('Name','OT 1,2 PSD','NumberTitle','on')
figure(10)
    subplot(2,1,1)
        plot(ff,h_Pot5x,ff,Pot5x),grid,zoom  %,ff,h_arsx,ff,h_psdx
        %plot(ff,Pot3x,ff,Pa1x),grid,zoom
        %title('Spectral Density OT1 and OT2 - Accel. in Y direction')
        title(['Power Spectral Density using Welchs method with window length
= ',num2str(window),' - OT5x '])
        xlabel('frequency, Hz')
        ylabel('dB/Hz')
        legend('Uncontrolled','Controlled')   %,'ARS','PSD'
        axis([0 100 -inf inf]);
    subplot(2,1,2)
        %plot(ff,Pot3x,ff,Pot3y),grid,zoom
        plot(ff,h_Pot5y,ff,Pot5y),grid,zoom %,ff,h_arsy,ff,h_psdy
        title('Power Spectral Density using Welchs method - OT5y ')
        xlabel('frequency, Hz')
```

```
        ylabel('dB/Hz')
        legend('Uncontrolled','Controlled')  % 'ARS,'PSD'
        axis([0 100 -inf inf]);
end
%-------------------------------------------------------------------------
if psd_plt_z==1;
figure(11)
    plot(ff,Pa1y,ff,Pa2y),grid,zoom
    title('Spectral Density a1y and a2y - Accel. in Z direction')
    xlabel('frequency, Hz')
    ylabel('dB/Hz')
    legend('accel 1','accel 2')
end
%-------------------------------------------------------------------------
if OT1_pos==1;
%figure('Name','OT 1','NumberTitle','on')
figure(12)
    subplot(2,1,1)
    plot(tt,ot1x),grid,zoom
     ylabel(['x pos, \mu','m'])
     %xlabel('x pos, micrometers')
     title('OT1 pos on detector vs. time')
     %axis([-1 1 -1 1]);
     %axis equal
     subplot(2,1,2)
    plot(tt,ot1y),grid,zoom
     ylabel(['y pos, \mu','m'])
     xlabel('time,sec')
     %legend('ot3y','ot3x')
     %axis([-1 1 -1 1]);
end
%-------------------------------------------------------------------------
if OT2_pos==1;
    meanstx = roundn(mean(ot2x(round(0.9/Ts):round(1.0/Ts))),-2);
    meansty = roundn(mean(ot2y(round(0.9/Ts):round(1.0/Ts))),-2);
    meanspx = roundn(mean(ot2x((round(1.03/Ts)):round(1.1/Ts))),-2);
    meanspy = roundn(mean(ot2y((round(1.03/Ts)):round(1.1/Ts))),-2);
%figure('Name','OT 2','NumberTitle','on')
figure(13)
    subplot(2,1,1)
    plot(tt,ot2x),grid,zoom
     ylabel(['x pos, \mu','m'])
     %xlabel('x pos, micrometers')
     title('OT2 pos on detector vs. time')
     %axis([-inf inf 50 75]);
     %axis equal
     %title(['Step Response: X Axis start : ',num2str(meanstx),'\mu, end :
',num2str(meanspx),...
      %      '\mu; Y Axis start : ',num2str(meansty),'\mu, end :
',num2str(meanspy),'\mu'])
     subplot(2,1,2)
    plot(tt,ot2y),grid,zoom
     ylabel(['y pos, \mu','m'])
     xlabel('time,sec')
     %legend('ot3y','ot3x')
```

```
     %axis([-1 1 -1 1]);

end
%-------------------------------------------------------------------------
if OT4_pos==1;
%figure('Name','OT 4','NumberTitle','on')
figure(14)
    subplot(2,1,1)
    plot(tt,ot4x/500),grid,zoom
     ylabel(['x pos, \mu','m'])
     %xlabel('x pos, micrometers')
     title('OT4 pos on detector vs. time')
     %axis([-1 1 -1 1]);
     %axis equal
     subplot(2,1,2)
    plot(tt,ot4y),grid,zoom
     ylabel(['y pos, \mu','m'])
     xlabel('time,sec')
     %legend('ot3y','ot3x')
     %axis([-1 1 -1 1]);
end
%-------------------------------------------------------------------------
if OT5_pos==1;
%figure('Name','OT 5','NumberTitle','on')
figure(15)
    subplot(2,1,1)
%     plot(tt,-ot5x/500),grid,zoom
plot(tt,ot5x),grid,zoom
    %title(['Target X pos vs. time - ','Std Dev: X Axis -
',num2str(xstdin),'\mum'])
     ylabel(['x pos, \mu','m'])
     %legend('ot5x','x_pred')
     %xlabel('x pos, micrometers')
     %title('OT5 pos on detector vs. time')
     %axis([-1 1 -1 1]);
     %axis equal
     subplot(2,1,2)
    plot(tt,ot5y),grid,zoom
    %title(['Target Y pos vs. time - ','Std Dev: Y Axis -
',num2str(ystdin),'\mum'])
     ylabel(['y pos, \mu','m'])
     xlabel('time,sec')
     %legend('ot5y','y_pred')
     %axis([-1 1 -1 1]);

end
%-------------------------------------------------------------------------
if OT_plot_compare==1;
%     diff_otx = sqrt(sum((ot5x-x_pred).^2))/length(ot5x);
%     diff_oty = sqrt(sum((ot5y-y_pred).^2))/length(ot5y);
     diff_otx = sqrt(mean((x_pred(1001:end)-ot5x(1001:end)).^2));
     diff_oty = sqrt(mean((y_pred(1001:end)-ot5y(1001:end)).^2));

figure(16)
    subplot(2,1,1)
```

```
    plot(tt,ot5x,tt,x_pred),grid,zoom
    legend('measured x','predicted x')
    ylabel(['\mu','m '])
    title(['Beam Position at Target Measured vs Predicted in \mu','m '])
    subplot(2,1,2)
    plot(tt,ot5y,tt,y_pred),grid,zoom
    title(['RMS: x pos = ',num2str(diff_otx,2),' \mum  ',...
        'y pos = ',num2str(diff_oty,2),' \mum'])
    legend('measured y','predicted y')
    ylabel(['\mu','m '])
    xlabel('time,sec')
end
%-------------------------------------------------------------------------
if test_inp==1;
figure(17)
    plot(tt,xtest,tt,ytest),grid,zoom
    %plot(tt,-100*IA_1_input,tt,ot5y),grid,zoom
    legend('xtest','ytest')
    xlabel('time,secs')
    ylabel('test input, 1000 \muvolts, and displacement, \mu m')
    title('input - output')
end
%-------------------------------------------------------------------------
if rot==1;
figure(18)
    plot(tt,pitch_rate,tt,roll_rate,tt,yaw_rate),grid,zoom
    legend('pitch rate','roll rate','yaw rate')
    xlabel('time,secs')
    ylabel('\murads/sec')
    title('Rotational Rates')
end
%-------------------------------------------------------------------------
if rotations==1;
figure(19),
    subplot(3,1,1)
    plot(tt,pitch_1,tt,pitch_2,tt,pitch_3),grid,zoom
    legend('pitch PSD','pitch integ ARS','pitch pred ARS')
    xlabel('time,secs')
    ylabel('\murads')
    title({'Plate Rotations','',[' ARS Rotation Measurement Error, RMS: Pitch
',...
            num2str(rot_error_pitch,2),' \murads, Roll
',num2str(rot_error_roll,2),...
            ' \murads, Yaw  ',num2str(rot_error_yaw,2),' \murads']});
        subplot(3,1,2)
    plot(tt,roll_1,tt,roll_2,tt,roll_3),grid,zoom
    legend('roll PSD','roll integ ARS','roll pred ARS')
    xlabel('time,secs')
    ylabel('\murads')
        subplot(3,1,3)
    plot(tt,yaw_1,tt,yaw_2,tt,yaw_3),grid,zoom
    legend('yaw PSD','yaw integ ARS','yaw pred ARS')
    xlabel('time,secs')
    ylabel('\murads')
end
```

```
%-------------------------------------------------------------------------
if rot_cal==1;
figure(20),title('Plate Rotation Rates')
    subplot(3,1,1)
    plot(tt,pitch_rate_calibrated),grid,zoom
    legend('pitch rate')
    xlabel('time,secs')
    ylabel('\murads/sec')
        subplot(3,1,2)
    plot(tt,roll_rate_calibrated),grid,zoom
    legend('roll rate')
    xlabel('time,secs')
    ylabel('\murads/sec')
        subplot(3,1,3)
    plot(tt,yaw_rate_calibrated),grid,zoom
    legend('yaw rate')
    xlabel('time,secs')
    ylabel('\murads/sec')
end
%-------------------------------------------------------------------------
if jitter==1,'Color',[0 0.502 0]
    figure(21)
    plot(tt,ot5j,tt,rmean_jitter),grid,legend('Jitter Angle')
    title(['Percent Improvement in Mean Jitter Angle =
',num2str(imp_mean_jitter,4),'%'...
        ' , ',num2str(Mean_jitter_after,4),' \murad'])
    xlabel('sec'),ylabel('\murad'),axis([-inf inf -inf inf]);
end
%-------------------------------------------------------------------------
if temp==1
    figure(22)
    plot(tt,temp1,tt,temp2),grid,legend('temp1','temp2')
    xlabel('sec'),ylabel('deg C')
end
if OT6_pos==1;
%figure('Name','OT 1','NumberTitle','on')
figure(23)
    subplot(2,1,1)
    plot(tt,ot6x),grid,zoom
     ylabel(['ot6x pos, \mu','m'])
     %xlabel('x pos, micrometers')
     title('OT6 pos on detector vs. time')
     %axis([-1 1 -1 1]);
     %axis equal
     subplot(2,1,2)
    plot(tt,ot6y),grid,zoom
     ylabel(['ot6y pos, \mu','m'])
     xlabel('time,sec')
     %legend('ot3y','ot3x')
     %axis([-1 1 -1 1]);
end

% %-------------------------------------------------------------------------
%Save Experimental Data
%savefile     = 1;   %   Set to one to save data
```

```
%Change this folder location to where you want the data saved to!
% c1= 'C:\Users\Trident\Documents\Moran\Experiments\14FEB\ex';
c1= 'F:\Date\16FEB\Moran\Experiments\14FEB\ex';


savefile = input('Do you wish to save this data? y/n [n]: ','s');
if savefile=='y';
    reply1 = input('input experiment number   ','s');
    reply2 = input('input run number   ','s');
    c2= reply1;  c3='_run'; c4=reply2;c5='.mat';c6='.fig';
    strsave = strcat(c1,c2,c3,c4,c5);
    if exist(strsave,'file')
        reply3 = input('THIS FILE EXISTS - OK TO OVERWRITE?  y/n [n]: ','s');
        if isempty(reply3);reply3='n';end
        if reply3 ~= 'y';
            beep
            reply1 = input('input experiment number   ','s');
            reply2 = input('input run number   ','s');
            c2= reply1;  c4=reply2;
            strsave = strcat(c1,c2,c3,c4,c5);
        end
    end
    save(strsave,...
        'tt','ot1y','ot1x','ot2y','ot2x','ot3y','ot3x','pitch_1',...
        'roll_1','yaw_1','pitch_2','roll_2',...
        'yaw_2','pitch_3','roll_3',...
        'yaw_3','ot5y','ot5x','ot5j','pitch_rate','roll_rate',...
        'yaw_rate','FSMAVx','FSMAVy','m1xc','m1yc','y_pred','x_pred',...
        'ff','h_Pot5x','h_Pot5y',...
        'Pot5x','Pot5y')

 end
% %-------------------------------------------------------------------------
%end
% save noise_floor2.mat;
%%

ot5r = (((ot5y-cal_ot5y*500).^2+(ot5x-cal_ot5x*(-500)).^2).^0.5); % miss dist
in um
ot5j = ot5r./dist_targ;  % jitter in urad
rmean_jitter=smooth(ot5j,300,'moving');


beep;
home;
```

APENDIX K: User Buffer Block (H_∞ Controller) Script

```
function y = buffer2(u)
persistent datt

N=1024;
if isempty(datt)
    datt=ones(N,1);
end
%Shift everything right 1 step
datt(2:N)=datt(1:(N-1));
%Insert new data
datt(1)=u;
y = datt;
```

APENDIX L: Frequency Identification and Controller Calculation Script

```
function [Ad1,Bd1,Cd1,Dd1,freqID]=sim_control_check3(LoMags,BufferData)
%#codegen
persistent cnt freqIDs

T=.001; % sample rate
N=1024; % buffer length
num_freqs=2; %Number of frequncies you wish to attenuate
cut_off=7;
res_freq=6;
%% Initialization
 if isempty(cnt)
     cnt=-N;
     freqIDs=zeros(1,num_freqs);
 end
%%Delays controller calculation until 2 second mark
%% 1st second nothing...2nd second uncontrolled

if sum(isinf(BufferData))<3 && cnt==N
    temp=zeros(N/2,2);
    temp2=zeros(num_freqs,2);
    temp3=zeros((N/2-cut_off),2);


    zetas=zeros(1,num_freqs);

    n_feval=0;
    ContIn=zeros(num_freqs,2);
    Con_A=zeros(4+4*num_freqs);
    Con_B=zeros((4+4*num_freqs),1);
    Con_C=zeros(1,(4+4*num_freqs));
    Con_D=0;

    %% Add Freqeuncies to Magnitudes

    temp(:,1)=(0:N/2-1)./(N*T); % Assigns frequency to first column
    temp(:,2)=BufferData(1:(N/2));
    temp3=temp(cut_off:size(temp,1),:);
    %% Calculate Threshold
    thresh=max(temp3(:,2))-12;
    temp3(:,2)=temp3(:,2)-thresh;
    I= temp3(:,2)>0 & temp3(:,1)<450; %Logical check of rows with magnitude
greater then 0
    temp3=temp3(I,:); %Remove all frequencies with negative magnitudes
    temp3(:,2)=temp3(:,2)+20.5;
    [Y,I]=sortrows(temp3,-2); %Sort Descending Magnitude
    temp3=temp3(I,:);

    for i=1:num_freqs
        temp2(i,:)=temp3(1,:);
        I=abs(temp3(:,1)-temp3(1,1))<3.5;
        temp3(I,2)=0;
```

```
        [Y,I]=sortrows(temp3,-2);
        temp3=temp3(I,:);
    end

    [Y,I]=sortrows(temp2,1); % negative sorts in ascending order
    temp2=temp2(I,:); %Reorder rows by frequency.



    for i=1:length(temp2)
        if temp2(i,2)~=0
            ContIn(i,2)=zetacalc(temp2(i,2));
        end
    end

    ContIn(:,1)=temp2(:,1);
    freqIDs=ContIn(:,1)';
    [Con_A,Con_B,Con_C,Con_D]=ControlDesign(ContIn);
    cnt=-N;
else
    Con_A=zeros(4+4*num_freqs);
    Con_B=zeros((4+4*num_freqs),1);
    Con_C=zeros(1,(4+4*num_freqs));
    Con_D=0;
end
[Ad1,Bd1,Cd1,Dd1]=tus(Con_A,Con_B,Con_C,Con_D,T);
cnt=cnt+1;
freqID=freqIDs;
end


function [zeta1]=zetacalc(gain)
z2=.06;

zeta1=10^((gain)/20)*z2;


end

function [ConA,ConB,ConC,ConD] = ControlDesign(zetas)
%CONTROLDESIGN intakes the calculated zetas (array size does not matter)
num=2;
%_f matricies are weight realizations
Af=zeros(2);
Bf=zeros(2,1);
Cf=zeros(1,2);
Df=zeros(1);
%Kinf_ matricies are Kinf Controller for given weight
KinfA=zeros(2);
KinfB=zeros(2,1);
KinfC=zeros(1,2);
KinfD=zeros(1);
%Con_ matricies are Final Control Matricies
```

```
ConA=zeros(4+num*4);
ConB=zeros(4+4*num,1);
ConC=zeros(1,4+4*num);
ConD=zeros(1);
%temp_ matricies are cascade of _f and Kinf_ matricies to then cascade into
%Con_ matricies
tempA=zeros(4);
tempB=zeros(4,1);
tempC=zeros(1,4);
tempD=zeros(1);
X=zeros(2);
Z=zeros(2);
zeta2=0.06;
%% Insert initial Ko Wo realization into controller. This is taken from
LoCalcs

ConA(1:4,1:4)=[-2.45088325420607,-43.4951925514125,-88.3266944027169,0;-
43.4951925514313,-809.100708908930,-
2436.36862875800,0;88.3266944027628,2436.36862875800,-
1802.56834487364,0;0,0,0,1.38777878078145e-17;];
ConB(1:4,1)=[0.295572591408228;4.31926829849558;-
3.99986581392468;0.500041104337736;];
ConC(1,1:4)=[0.295572591408326,4.31926829849557,3.99986581392468,0.0554612297
986527;];
for i=1:size(zetas,1)
    if zetas(i,2)~=0
        omegaN1=2*pi*zetas(i,1);
        Af=[-2*zeta2*omegaN1 1; -(omegaN1^2) 0];
        Bf=[2*omegaN1*(zetas(i,2)-zeta2); 0];
        Cf=[1 0];
        Df=1;
        X=[-(zetas(i,2)+zeta2-((zetas(i,2)-
zeta2)^2+(zetas(i,2)+zeta2)^2)^(1/2))/(2*zetas(i,1)*(zetas(i,2)-zeta2)^2) 0;
0 -(zetas(i,2)+zeta2-((zetas(i,2)-
zeta2)^2+(zetas(i,2)+zeta2)^2)^(1/2))/(2*zetas(i,1)^3*(zetas(i,2)-zeta2)^2)];
        Z=[-2*zetas(i,1)*(zetas(i,2)+zeta2-((zetas(i,2)-
zeta2)^2+(zetas(i,2)+zeta2)^2)^(1/2)) 0; 0 -2*zetas(i,1)^3*(zetas(i,2)+zeta2-
((zetas(i,2)-zeta2)^2+(zetas(i,2) + zeta2)^2)^(1/2))];
        eigZX=(zetas(i,2)+zeta2-((zetas(i,2)-
zeta2)^2+(zetas(i,2)+zeta2)^2)^(1/2))^2/(zetas(i,2)-zeta2)^2;
        gam=1.05*sqrt(1+eigZX);
        del=gam^2*(((1-gam^2)*eye(2)-Z*X)^-1)*Z'*Cf';
        S=eye(size(Df'*Df,1))+Df'*Df;
        F=-inv(S)*(Df'*Cf+Bf'*X);
        KinfA=Af+Bf*F+del*(Cf+Df*F);
        KinfB=-del;
        KinfC=Bf'*X;
        KinfD=Df';
        tempA=[Af Bf*KinfC; zeros(2) KinfA];
        tempB=[Bf*KinfD;KinfB];
        tempC=[Cf Df*KinfC];
        tempD=[Df*KinfD];
        ConA(1:4+4*i,1:4+4*i)=[tempA tempB*ConC(1,1:4*i); zeros(4*i,4)
ConA(1:4*i,1:4*i)];
        ConB(1:4+4*i,1)=[tempB*ConD; ConB(1:4*i,1)];
```

```
        ConC(1,1:4+4*i)=[tempC tempD*ConC(1,1:i*4)];
        ConD=[tempD*ConD];
    end
end
end


function [Adt,Bdt,Cdt,Ddt]=tus(A,B,C,D,T)

Al=A*(T/2);
Bl=B*(T/2);
A2=2*(inv(eye(size(Al))-Al))-eye(size(Al));
B2=2*(inv(eye(size(Al))-Al))*Bl;
C2=(inv(eye(size(Al))-Al));
D2=(inv(eye(size(Al))-Al))*Bl;
Adt=A2;
Bdt=B2;
Cdt=C*C2;
Ddt=(D+C*D2);
end
```

APENDIX M: Discrete State Space Controller Script

```
function u  = FSM_control(Ad, Bd, Cd, Dd, signal,freqID)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
persistent Ad1 Bd1 Cd1 Dd1 Xk Xk2 t run2 IDfreq

N=1024;
num=2;

% Initialize all components
% A low pass filter is created until first controller calculate
% Can run LPFd to calculate it.
if isempty(t)
    t=0;
    Ad1=zeros((4+4*num));
    Bd1=zeros((4+4*num),1);
    Cd1=zeros(1,(4+4*num));
    Xk=zeros((4+4*num),1);
    Xk2=zeros((4+4*num),1);
    Ad1(1:4,1:4)=[-0.673287251524460,-0.361701453252854,0.0400149220625872,-
0.0687677369019813;0.554669443529610,0.0438669327677170,-0.600155994264337,-
6.40978365414666e-05;0.112600151997707,0.211909231103608,-
0.328849476193162,0.210797334465462;0,0,0,-0.310344827586207;];
    Bd1(1:4,1)=[-0.550141895215851;-
0.000512782692331733;1.68637867572370;5.51724137931034;];

Cd1(1,1:4)=[0.00736032628285046,0.00817684952879757,0.00218104140642532,0.001
68887485395321;];
    Dd1=[0.0135102788316257;];
    run2=0;
    IDfreq=zeros(1,num);

end

if t==2*N && run2==0
    run2=1;
    t=N;
end

if t==N & run2==1 & sum(((IDfreq-freqID).^2))^.5>5.5
%     Xk=Xk+Cd'*(((Cd1-Cd)*Xk)/(Cd*Cd')); %Shift new state so output is
continuous
    Xk=zeros((4+4*num),1);

    Ad1=Ad;
    Bd1=Bd;
    Cd1=Cd;
    Dd1=Dd;
    t=-1024;
    IDfreq=freqID;
end

Xk2=Ad1*Xk+Bd1*signal;
```

```
u=Cd1*Xk+Dd1*signal;
Xk=Xk2;

t=t+1;



end
```